НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»

ФАКУЛЬТЕТ ІНФОРМАТИКИ ТА ОБЧИСЛЮВАЛЬНОЇ ТЕХНІКИ

Кафедра автоматизованих систем обробки інформації і управління

«На правах рукопису»

УДК _____

«До захисту допущено»
В.о. завідувача кафедри
_____ *О.А.Павлов*
(підпис) (ініціали, прізвище)
“___”_____2019 р.

Магістерська дисертація

зі спеціальності *121 «Інженерія програмного забезпечення»*

на тему: «*Математичне та програмне забезпечення визначення автора художнього тексту*»

Виконав: студент VI курсу, групи *ІП-82мп*

Храпов Олег Олегович _____
(прізвище, ім’я, по батькові) (підпис)

Науковий керівник *доц., к.т.н. Фіногенов О.Д.* _____
(посада, науковий ступінь, , вчене звання, прізвище та ініціали) (підпис)

Консультант *доц., к.т.н. Ліщук К.І.* _____
(посада, науковий ступінь, вчене звання, прізвище та ініціали) (підпис)

Рецензент *доц. каф. вир. прил. ПБФ, доц., к.н. Філіппова М. В.* _____
(посада, науковий ступінь, вчене звання, прізвище та ініціали) (підпис)

Засвідчую, що у цій магістерській дисертації немає запозичень з праць інших авторів без відповідних посилань.

Студент _____
(підпис)

Київ – 2019 року

Національний технічний університет України
«Київський політехнічний інститут
імені Ігоря Сікорського»

Факультет *інформатики та обчислювальної техніки*
(повна назва)

Кафедра *автоматизованих систем обробки інформації та управління*
(повна назва)

Рівень вищої освіти *другий (магістерський) за освітньо-професійною програмою*

Спеціальність *121 «Інженерія програмного забезпечення»*
(код і назва)

ЗАТВЕРДЖУЮ
В.о. завідувача кафедри
_____ *О.А. Павлов*
(підпис) (ініціали, прізвище)
«___» _____ 20*19* р.

ЗАВДАННЯ
на магістерську дисертацію студенту
Храпову Олегу Олеговичу
(прізвище, ім'я, по батькові)

1. Тема дисертації Математичне та програмне забезпечення визначення авторства художнього тексту

науковий керівник дисертації *доц., к.т.н.* Фіногенов О.Д
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом по університету від "____" _____ 20__ р. № _____

2. Строк подання студентом дисертації "____" _____ 20__ р.

3. Об'єкт дослідження Стилістичні особливості письмового стилю автора. Визначення автора тексту. Процес визначення характеристик та класифікації

4. Предмет дослідження Лексичні та синтаксичні ознаки письмового стилю автора художнього тексту та алгоритм класифікації текстових даних

5. Перелік завдань, які потрібно розробити дослідити відомі на даний час

методи та підходи визначення стилю автора тексту. Збір навчальних даних, розробка класифікатору на основі обраних моделей. Тестування та аналіз ефективності використаних моделей

6. Перелік графічного матеріалу
   Схема роботи програмного забезпечення
   Архітектура класифікатора

7. Орієнтовний перелік публікацій Визначення автора тексту з використанням штучних нейронних мереж. Визначення статі автора короткого тексту методами машинного навчання

8. Консультанти розділів дисертації

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв

9. Дата видачі завдання “ *01* ” *вересня* 20 *19* р

Календарний план

№ з/п	Назва етапів виконання магістерської дисертації	Строк виконання етапів магістерської дисертації	Примітка
1	*Формалізація результатів огляду систем визначення автора тексту*		
2	*Порівняльний аналіз існуючих методів визначення стилю автора*		
3	*Постановка та формалізація математичної моделі задачі*		
4	*Розробка архітектури системи*		
5	*Розробка програмного забезпечення*		
7	*Проведення експериментальних досліджень використаних моделей*		
8	*Оформлення документації*		
9	*Подання роботи на попередній захист*	*05.12.2019*	
10	*Подання роботи на основний захист*	*16.12.2019*	

Студент _____ (підпис) _____ (ініціали, прізвище)

Науковий керівник _____ (підпис) _____ (ініціали, прізвище)

РЕФЕРАТ

Актуальність теми: виконання класифікації художніх текстів людиною (наприклад редактором) на наявність порушення авторських прав або при перевірці плагіату вимагає багато часу. Вирішенням цієї проблеми є автоматизація пошуку автора на основі аналізу стилістичних особливостей письма автора. За рахунок використання машинного навчання можна створити автоматичний класифікатор, який дозволить покращити точність класифікації порівняно з ручною.

Мета дослідження: розробити моделі, що забезпечують розпізнавання лексичних та стилістичних ознак автора художнього тексту.

Для реалізації поставленої мети були сформульовані **наступні завдання**:

- дослідити відомі на даний час методи та підходи визначення стилю автора тексту;
- збір навчальних даних, розробка класифікатору на основі обраних моделей;
- тестування та аналіз ефективності використаних моделей;
- визначення подальшого напряму досліджень.

Об’єкт дослідження: стилістичні особливості письмового стилю автора. Визначення автора тексту. Процес визначення характеристик та класифікації художніх текстів на українській мові.

Предмет дослідження: лексичні та синтаксичні ознаки письмового стилю автора художнього тексту та алгоритм класифікації текстових даних.

Методи дослідження: для розв’язання даної задачі використовувались лексичні та синтаксичні ознаки тексту, нейроні мережі, алгоритм зворотнього поширення помилки.

Наукова новизна: найбільш суттєвими науковим результатами магістерської дисертації є розроблена модель визначення автора тексту, проаналізувавши його стилю письма, за допомогою моделі пунктуаційних та функціональних слів текстів написаних на українській мові.

Практичне значення отриманих результатів визначається тим, що запропонований та розроблений алгоритм навчання та визначення дозволяє досягти точності визначення автора тексту в 80%, що дозволяє використовувати його для перевірки текстів у визначенні авторських прав або при перевірці документів на плагіат.

Зв'язок роботи з науковими програмами, планами, темами: робота виконувалась на кафедрі автоматизованих систем обробки інформації та управління Національного технічного університету України «Київський політехнічний інститут ім. Ігоря Сікорського» в рамках теми «Математичні моделі та технології в СППР». Державний реєстраційний номер 0117U000914

Апробація: результати викладалися на предзахисті роботи на кафедрі АСОІУ.

Публікації: Визначення автора тексту з використанням штучних нейронних мереж. Міжнародний електронний науковий журнал - 2019. - №12. - URL: https://nauka-online.com/ua/publications/informatsionnye-tehnologii/ 2019/ 12/ viznachennya-avtora-tekstu-z-vikoristannyam-ann/.

Визначення статі автора короткого тексту методами машинного навчання. Міжнародний електронний науковий журнал - 2019. - №11. – URL: https://nauka-online.com/ua/publications/tehnicheskie-nauki/2019/11/opredelenie-pola-avtora-korotkogo-teksta-metodami-mashinnogo-obucheniya/.

Ключові слова: СТИЛІСТИЧНІ ОЗНАКИ, МАШИННЕ НАВЧАННЯ, КЛАСИФІКАЦІЯ, ВИЗНАЧЕННЯ АВТОРА.

ABSTRACT

Topicality of the topic: it takes a long time to classify artistic texts by a person (for example, an editor) for copyright infringement or for plagiarism verification. The solution to this problem is to automate the search for the author based on an analysis of the stylistic features of the author's letter. By using machine learning, you can create an automatic classifier that will improve the accuracy of classification compared to manual.

Purpose of the study: to develop models that provide recognition of the lexical and stylistic features of the author of the artistic text.

To achieve this goal, the following tasks were formulated:

– explore currently known methods and approaches for determining the style of the author;

– collection of training data, development of a classifier based on selected models;

– testing and analyzing the efficiency of the models used;

– defining further direction of research.

Object of study: stylistic features of the author's writing style. Definition of the author of the text. The process of determining the characteristics and classification of artistic texts in the Ukrainian language.

Subject of study: lexical and syntactic features of the written style of the author of the artistic text and algorithm for classification of textual data.

Research Methods: To solve this problem, we used lexical and syntactic features of the text, neural networks, the algorithm of reverse error propagation.

Scientific novelty: the most significant scientific result of the master's thesis is the developed model of determining the author of the text, analyzing his style of

writing, using a model of punctuation and functional words of texts written in Ukrainian.

The practical significance of the obtained results is determined by the fact that the proposed and developed algorithm of training and definition allows to achieve the accuracy of the definition of the author of the text in 80%, which allows to use it for verification of texts in the definition of copyright or when checking documents for plagiarism.

Relationship with working with scientific programs, plans, topics: work was performed at the Department of Automated Information Processing and Management Systems of the National Technical University of Ukraine «Kyiv Polytechnic Institute. Igor Sikorsky ”within the topic“ Mathematical Models and Technologies in DSS ”. State Registration Number 0117U000914

Testing: the results were taught in pre-protection work at the department of ASOIU.

Publications: Identification of the author of the text using artificial neural networks. International Electronic Scientific Journal - 2019. - №12. – URL: https://nauka-online.com/en/publications/informatsionnye-tehnologii/ 2019/12 / viznachennya-avtora-tekstu-z-vikoristannyam-ann /.

Defining the gender of the author of the short text by machine learning methods. International Electronic Scientific Journal - 2019. - №11. – URL: https://nauka-online.com/en/publications/tehnicheskie-nauki/2019/11/opredelenie-pola-avtora-korotkogo-teksta-metodami-mashinnogo-obucheniya/.

Keywords: STYLISTIC SIGNS, MACHINE LEARNING, CLASSIFICATION, AUTHOR DEFINITION.

ЗМІСТ

ВСТУП

Атрибуція авторства також відома як авторське визнання, авторська перевірка. Він використовується для ідентифікації авторів або письменників з книг, електронних листів, блогів і наукових статей. Атрибуція авторства використовує стиліометрію або текстометрію. Головне завдання атрибуції - визначити відповідну характеристику документів, що відображають стиль написання авторів. Авторство атрибуції часто плутають з авторським профілюванням або профілюванням лінгвістики. Профілювання автора стосується визначення особливостей автора, таких як гендер, ім'я, географічне походження, риси особистості тощо, але зазвичай не намагається ідентифікувати конкретну людину. Її мета полягає в тому, щоб зменшити простір пошуку так, щоб він не ставав проблемою.

Визначення та приписування авторства різних текстових фрагментів може бути корисним для різних завдань та областей, включаючи бібліометрію, пошук інформації та виявлення плагіату. Було розроблено концепцію розширення процесу пошуку, включивши інформацію про авторство, щоб дозволити ідентифікувати текстові фрагменти, написані певним автором, а також нею представлено нову функціональну класифікацію наукових документів, які фіксують розподіл стилометричних особливостей по всьому документу і передбачають відповідність кількості авторів.

Більшість досліджень в стилометрії обмежені або з точки зору використовуваних ознак, або з використанням нереалістичних наборів даних. Більше того, більшість відомих продуктів виявлення автора, таких як Turnitin [1], Viper [2], Paper Rater [3], не мають функціональності для аналізу та порівняння стилів написання. в документах, що подаються до них. Тільки Grammarly [4] певною мірою аналізує текст, але тоді акцент робиться лише на виявленні граматичних помилок, а не на стильометриці.

Завдання визначення або перевірки авторства анонімного тексту, що ґрунтується виключно на внутрішніх свідченнях, є дуже старим, що датується, принаймні, середньовічною схоластикою, для якої надійне привнесення даного тексту відомому старовинному авторитету було важливим для

визначення правдивості тексту. Зовсім недавно ця проблема атрибуції авторства набула більшої популярності завдяки новим застосуванням у судовому аналізі, гуманітарній науці та електронній комерції, а також розробці обчислювальних методів вирішення проблеми.

У найпростішій формі задачі наводяться приклади написання ряду кандидатів і просять визначити, який з них є автором даного анонімного тексту. Проблема авторства в цій прямолінійній формі відповідає стандартній сучасній парадигмі тексту проблеми категоризації. Компоненти систем категоризації текстів тепер досить добре зрозумілі: документи представлені як чисельні вектори, які фіксують статистику потенційно релевантних особливостей тексту, а методи машинного навчання використовуються для пошуку класифікаторів, які відокремлюють документи, що належать до різних класів.

Проте, проблеми авторства в реальному житті рідко бувають настільки елегантними, як прості проблеми категоризації тексту, в яких ми маємо невеликий закритий набір кандидатів і по суті необмежений текст для кожного з них. Існує ряд різновидів проблем атрибуції, які не відповідають даному ідеалу.

Метою роботи є вивчення історії методів, використаних для сценарію атрибуції основного авторства, і створити новий механізм атрибуції тексту на основі декількох вже відомих підходів та створити програмний продукт, який за допомогою нейронної мережі буде визначати автора ще невідомого програмі тексту.

1 ОСНОВИ ТЕКСТОВОЇ КЛАСИФІКАЦІЇ ТА ОГЛЯД ІСНУЮЧИХ РІШЕНЬ

1.1 Історія методів атрибуції

Більшість дослідників розглядають проблему визначення автора тексту, як проблему вилучення функцій і отримання даних. На інтуїтивному і неформальному рівні майже кожен знайомий з цим процесом. Підгрупи людей мають стереотипні звички мови.

Таким чином, більшість дослідників виходять з того, що у людей є характерна модель використання мови, свого роду «авторський відбиток», який можна виявити в їх працях. Ван Халтер [5] у свій праці дійшов до висновку, що кожна людина володіє специфічним набором вимірюваних рис, які можна використовувати для унікальної ідентифікації даного автора.

Протягом останнього століття і більше було застосовано велику різноманітність методів авторства з різними видами атрибуції. Підхід до машинного навчання, в якому сучасні методи машинного навчання застосовуються до наборів навчальних документів для побудови класифікаторів, які можуть бути застосовані до нових анонімних документів.

Основна ідея авторської атрибуції в статистичному або обчислювальному відношенні полягає в тому, що, вимірюючи деякі текстові особливості, ми можемо розрізняти тексти, написані різними авторами. Перші спроби дати кількісну оцінку стилю письма сходять до 19 століття, коли Менденхолл (1887) досліджував п'єси Шекспіра, а в першій половині 20 століття - Йоль (1938; 1944) і Ціпфа (статистичні дослідження в першій половині 20 століття). 1932). Пізніше докладне дослідження Мостеллер і Уоллеса (1964) про авторство «Документів федералістів» (серія з 146 політичних есе, написаних Джоном Джеєм, Олександром Гамільтоном і Джеймсом Медісон, дванадцять з яких стверджували і Гамільтон, і Медісон) була, безсумнівно, найвпливовішою роботою в атрибуції авторства. Їх метод був заснований на Байєсовому статистичному аналізі частот невеликого набору загальних слів і дозволив отримати значні результати відмінності між кандидатами в автори.

По суті, робота Мостеллер і Уоллеса (1964) ініціювала нетрадиційні авторські дослідження авторства, на відміну від традиційних людських експертних методів. З тих пір і до кінця 1990-х років в дослідженні авторства домінували спроби визначити особливості кількісного визначення стилю письма - напрямок досліджень, відоме як «стильометрія» (Holmes, 1994; Holmes, 1998).

Отже, було запропоновано велику різноманітність заходів, включаючи довжину пропозиції, довжину слова, частоти слів, частоти символів і функції збагачення словникового запасу. У деяких випадках були досягнуті вражаючі попередні результати і багато хто думав, що вирішення цієї проблеми було занадто близько. Найбільш характерним прикладом є метод CUSUM (або QSUM) (Morton & Michealson, 1990), який набув розголосу і був прийнятий в суді в якості експертного докази. Проте, дослідницьке співтовариство піддало його серйозній критиці і визнало в цілому ненадійним (Holmes & Tweedie, 1995). Власне, головною проблемою того раннього періоду була відсутність об'єктивної оцінки запропонованих методів. У більшості випадків полігоном були літературні твори невідомого або оспорюваного авторства, тому оцінка точності атрибуції була неможлива. Основними методологічними обмеженнями цього періоду щодо процедури оцінки були такі:

– текстові дані були занадто довгими (зазвичай включає цілі книги) і, ймовірно, не були стилістично однорідними;

– кількість кандидатів в автори було занадто мало (зазвичай 2 або 3);

– оціночні тіла не контролювалися по темі;

– оцінка запропонованих методів була в основному інтуїтивною (зазвичай ґрунтувалася на суб'єктивному візуальному огляді діаграм розсіювання);

– порівняння різних методів було ускладнено через відсутність відповідних контрольних даних.

З кінця 1990-х років в дослідженнях авторства авторства відбулися зміни. Величезна кількість електронних текстів, доступних через інтернет-ЗМІ

(електронні листи, блоги, онлайн-форуми і т.д.), збільшилася потреба в ефективній обробці цієї інформації. Цей факт справив значний вплив на наукові області, такі як пошук інформації, машинне навчання і обробка природної мови. Розвиток цих областей вплинуло на технологію авторства, як описано нижче:

– інформаційно-пошукові дослідження розробили ефективні методи для подання та класифікації великих обсягів тексту;

– стали доступні потужні алгоритми машинного навчання для обробки багатовимірних і розріджених даних, що дозволяє отримати більш виразні уявлення. Крім того, були розроблені стандартні методології оцінки для порівняння різних підходів на одних і тих же контрольних даних;

– дослідники розробили інструменти, здатні ефективно аналізувати текст і пропонувати нові форми заходів для подання стилю (наприклад, функції на основі синтаксису);

Отже, (приблизно) останнє десятиліття можна розглядати як нову еру технології аналізу авторства, в якій на цей раз переважають зусилля з розробки практичних додатків, що працюють з реальними текстами (наприклад, електронна пошта, блоги, повідомлення онлайн-форуму, вихідний код). і т. д.) замість вирішення спірних літературних питань. В даний час особлива увага приділяється об'єктивній оцінці пропонованих методів, а також порівняно різних методів, заснованих на загальних еталонних текстах (Juola, 2004).

У типовій задачі визначення авторства, тексту невідомого авторства присвоюються один кандидат автора, з огляду на безліч авторів-кандидатів, для яких текстові зразки безперечного авторства доступні. З точки зору машинного навчання, це можна розглядати як завдання класифікації тексту по одній мітці.

Оскільки кожна людина повинна вивчати «мову» сама по собі, і досвід вивчення мови різниться, то і мову, яку вивчає, відрізняється в мікро-аспектах. Існують також вагомі практичні причини вважати, що такі «відбитки пальців» можуть бути більш складними, ніж проста одномірна статистика, така як середня довжина слова або розмір словника.

Ключовим досягненням в дослідженнях стала розробка багатовимірних методів, налаштованих на функції розподілу замість простої наявності або відсутності функцій. Наприклад, замість того, щоб дивитися на конкретні слова, можна зосередитися на таких властивостях, як середня довжина слова або багатство словникового запасу. Навідміну від конкретних слів, ці властивості завжди доступні. Аналогічним чином, використання декількох функцій може привести до поліпшення порівняно з одновимірними функціями, оскільки вони надають більше інформації в цілому. Крім того, ці поліпшені функції можуть бути більш надійними, тому що вони менш сприйнятливі до прямих маніпуляцій.

1.2 Методологічні питання авторської атрибуції

У деяких областях (наприклад, при застосуванні авторства до юридичних питань), проблема точності є вирішальною. Без добре встановленої і задокументованої причини вважати, що аналіз є точним, він навіть не є допустимим доказом. Питання точності, є найбільш важливою проблемою, що стоїть сьогодні перед визначенням стильометрії.

Існує три основних взаємопов'язаних аспекти точності стильометрії:

– по-перше, це внутрішня точність самих методів. Існує майже необмежена кількість методів, існуючих для визначення авторства;

– проблеми жанру і розміру корпусу. Для літературних досліджень один автор зможе отримати кілька сотень тисяч слів для побудови навчальної вибірки, але, наприклад, розміри вибірки, доступні в типовому судовому розслідуванні, можуть становити всього кілька сотень або тисяч слів;

– легкість написання комп'ютерних програм для реалізації будь-яких знайдених методів забезпечує шлях до необгрунтованих заяв про точність. Якщо методи точні, цього все ж може бути недостатньо для наукового дослідження.

Результи програми не є задовільним поясненням для багатьох цілей (це також є одним з основних недоліків штучних нейронних мереж в якості осіб, які приймають рішення, і однією з причин того, що експертні системи зазвичай

зобов'язані давати пояснення причини, що лежать в основі їх рішень). Дослідник може не хотіти приймати рішення, ґрунтуючись тільки на результаті програми, в той час як дослідник, який цікавиться гендерними відмінностями в письмовій формі, може більше цікавитися причинами відмінностей, а не самими відмінностями.

Для аналізу повинні використовуватися тільки ті функції, за які автор несе пряму відповідальність. На практиці може бути дуже важко визначити, які функції належать автору. Більшість опублікованих робіт є продуктом багатьох рук, в тому числі автора, редактора, складача.

Навіть з «чистими» текстами присутність сторонніх (неавторизованих) матеріалів може бути проблемою. Також багато питанн може виникнути з цитатами. Хоча автор вибирає цитати для включення, і вони можуть спотворювати статистику. У крайніх випадках автор може навіть не помітити цитати або не знати, що його фрази запозичені в улюбленого автора. Для отримання максимально чистих зразків слід виключити всі сторонні матеріали, які не були отримані від автора. Завдання, що вимагає особливої обережності і знань з боку дослідника.

1.3 Лінгвістичні особливості тексту

Стилометричні характеристики. У дослідженнях авторства були запропоновані таксономії функцій для кількісної оцінки стилю письма, так званих маркерів стилю, під різними ярликами і критеріями. Поточний огляд можливостей представлення тексту в стилістичних цілях в основному сфокусований на обчислювальних вимогах для їх вимірювання. По-перше, лексичні і символьні особливості розглядають текст як просту послідовність слів-символів або символів, відповідно. Зверніть увагу, що, хоча лексичні особливості більш складні, ніж характерні риси. Потім синтаксичні та семантичні функції вимагають більш глибокого лінгвістичного аналізу, в той час як специфічні для додатка функції можуть бути визначені тільки в певних текстових доменах або мовами. Крім того, обговорюються різні способи

вибору і вилучення ознак для формування найбільш підходящого набору функцій для конкретного корпусу.

Лексичні особливості. Простий і природний спосіб перегляду тексту - це послідовність токенів, згрупованих в пропозиції, кожен з яких відповідає слову, числу або знаку пунктуації. Найперші спроби приписати авторство були засновані на простих заходах, таких як кількість пропозицій і кількість слів. Істотною перевагою таких функцій є те, що вони можуть застосовуватися до будь-якої мови і будь-якого корпусу без додаткових вимог, крім наявності токенізатора (тобто інструменту для сегментування тексту в токени).

Функції збагачення словникового запасу - це спроби кількісно оцінити різноманітність словникового запасу тексту. Типовими прикладами є відношення типу токенів V / N, де V - об'єм словника (унікальних токенів), а N - загальна кількість токенів тексту. Розмір словника сильно залежить від довжини тексту.

Найпростіший підхід до представлення текстів - це вектори частот слів. Переважна більшість авторських досліджень авторства засновані на лексичних особливостях для подання стилю. Це також традиційне текстове представлення з перечнем слів, за яким слідують дослідники в тематичної класифікації тексту. Таким чином, текст розглядається як набір слів, кожне з яких має частоту без урахування контекстної інформації. Проте, існує значна різниця в класифікації тексту на основі стилю: найбільш поширені слова (прийменники, займенники і т. д.). Зверніть увагу, що такі слова зазвичай виключаються з набору функцій методів класифікації тексту на основі теми, оскільки вони не несуть семантичної інформації, і їх називають функціональними словами. Як наслідок, класифікація тексту на основі стилю з використанням лексичних функцій вимагає набагато меншої розмірності в порівнянні з класифікацією тексту на основі теми. Функціональні слова використовуються авторами в значній мірі несвідомо і не залежать від теми. Таким чином, вони можуть захопити чисто стилістичний вибір авторів з різних тем.

Простий і дуже успішний метод визначення лексичного набору функцій для атрибуції авторства полягає в тому, щоб знайти часто використовувані слова з доступного корпусу. Потім необхідно прийняти рішення про кількість часто використовуваних слів, які будуть використовуватися в якості функцій. Однак наявність потужних алгоритмів машинного навчання, здатних працювати з тисячами функцій, таких як машини опорних векторів, дозволило дослідникам збільшити розмір набору функцій цього методу.

Синтаксичні особливості. Більш складним способом представлення тексту є використання синтаксичної інформації. Ідея полягає в тому, що автори схильні використовувати подібні синтаксичні патерни неусвідомлено. Тому синтаксична інформація вважається більш надійним авторським відбитком в порівнянні з лексичної інформацією. Одна з причин, по якій функціональні слова працюють добре, полягає в тому, що вони не залежать від теми.

Функціональні слова мають тенденцію бути семантично знебарвленими. Вони описують відносини між змістовними словами, свобода варіацій і особистий вибір в статистиці конкретних слів. Вони також можуть відображати більш широкий синонімічний характер, наприклад, між активними і пасивними конструкціями, використанням риторичних запитань проти простих декларативних виразів або використанням з'єднань замість ряду окремих тверджень. Іншими словами, кращі синтаксичні конструкції людини також можуть бути ознаками його авторства. Один простий спосіб зрозуміти, помітити відповідні документи для частини мови (POS). Пунктуація може бути іншим легко доступним джерелом такої інформації [1]. Недоліком такої обробки, особливо для POS-тегів, є внесення помилок в саму обробку; Система, яка не може розрізнити скорочуються апострофи і закриті одинарні лапки або може позначати тільки з точністю до 90%, буде містити абсолютно різні синтаксичні конструкції. Альтернативний підхід, який об'єднує лексичну і синтаксичну інформацію, - це використання слів N-грам (біграми, триграми і т. д.).

Семантичні особливості. Більш детальний аналіз тексту потрібен для виділення стильометричних ознак, тим менш точними виходять вимірювання. Інструменти можуть бути успішно застосовані до завдань низького рівня, таким як розбиття на теги POS, фрагментація тексту, частковий аналіз, тому відповідні функції будуть вимірюватися точно, а шум у відповідних наборах даних залишається низьким. З іншого боку, більш складні завдання, такі як переклад тексту синтаксичний аналіз, семантичний аналіз або прагматичний аналіз, ще не можуть бути адекватно виконані сучасною технологією для необмеженого тексту.

Гамон [7] використовував інструмент, здатний створювати графи семантичних залежностей, але він не надав інформацію про точність цього інструменту. Потім були вилучені два види інформації: бінарні семантичні ознаки і відносини семантичної модифікації. Перше стосувалося кількості іменників, часу і аспекту дієслів і т. д. Друге описує синтаксичні і семантичні відносини між вузлом графа і його точками. Представлені результати показали, що семантична інформація в поєднанні з лексичної і синтаксичної інформацією підвищує точність класифікації.

McCarthy, Lewis, Dufty і McNamara [8] описали інший підхід до вилучення семантичних показників. Грунтуючись на WordNet [9], вони оцінили інформацію про синоніми та гіпероніми слів, а також про ідентифікацію причинних дієслів. Крім того, вони застосували прихований семантичний аналіз по лексичним особливостям, щоб автоматично визначати семантичні подібності між словами. Тим не менш, не було докладного опису ознак, і процедура оцінки не прояснила внесок семантичної інформації в класифікаційну модель.

1.4 Атрибуційний аналіз

Ключовим фактором, що впливає на вибір методу аналізу, є розуміння вимог остаточної відповіді. Важливо не просто представити результати, але і пояснити їх з точки зору основних аспектів документів.

До інших важливих факторів належать такі питання, як кількість і тип доступних навчальних матеріалів. Наприклад, різниця між контрольованими і

неконтрольованими методами застосовується тут, як і скрізь в машинному навчанні. Контрольовані методи вимагають апріорного знання класу міток, часто у вигляді зразків документів безперечного авторства. Неконтрольовані методи більше підходять для дослідження даних без попередньої інформації. Ці аналітичні методи будуть розглянуті окремо.

Вектори простору. Кількісна оцінка об'єктів, в свою чергу, створить багатовимірний простір документів з набором функцій кожного документа, що визначає вектор або точку в цьому просторі. Є дві основні проблеми з цим. Перша проблема візуалізації п‘ятимірного простору, а друга проблема незалежності різних вимірів. Наприклад, документи, написані від першої особи, часто мають високу частоту займенників першої людини, таких як «я». Вони також можуть мати високу частоту дієслів першої особи.

Вимірювання частоти не є незалежними, так як насправді вони були всього лише вимірами ступеня експозиції від першої особи в письмовій формі, тобто кореляція серед цих частот буде позитивною.

Щоб вирішити цю проблему, дослідники [6, 7, 8] часто використовують аналіз головних компонентів (PCA. Технічно, PCA - це вектори коваріаційної матриці серед безлічі ознак. Неформально PCA визначає менший набір (ортогональних, незалежних) базових векторів, які описують як можна більшу частину варіацій у вихідному наборі даних. Зокрема, два основних компоненти описують вихідний набір даних в двовимірному просторі, зберігаючи в максимально можливій мірі схожість між окремими елементами даних.

Кластерний аналіз. Даний аналіз припускає наявність відстані між парами документів, яка вимірюється безпосередньо або виводиться з метрики, застосованої до векторного простору. В обох випадках аналіз кластера виконується шляхом угруповання найближчої пари елементів в кластер, а потім заміни цієї пари елементів новим елементом, що представляє сам кластер. На кожному етапі кількість аналізованих елементів зменшується на одиницю, поки всі елементи не будуть об'єднані в один кластер. Результат такого аналізу зазвичай відображається у вигляді кластерної діаграми,

кореневого дерева з бінарним розгалуженням. Висота кожного внутрішнього вузла представляє відстань, на якому була знайдена найближча пара, а дочірні елементи цього вузла представляють два елементи, з'єднаних на цьому кроці.

1.5 Класифікаційні алгоритми

Первинний аналіз предметної області дає розуміння, що дана задача відноситься до категорії класифікації текстів на основі виділених з них ознак. Операцію лінійної класифікації для двох класів можна представити як відображення об’єктів в багатовимірному просторі на гіперплощину, в якій об’єкти, що попали по одну сторону розділяючої лінії відносяться до умовно першого класу, а об’єкти по іншу - до другого класу.

Лінійний дискримінантний аналіз (LDA). Використовується, щоб знайти лінійну комбінацію функцій, що характеризує або розділяє два або більше класів об'єктів або подій. LDA є параметричним підходом в піднаглядний техніці навчання. Спочатку вона була використана для зменшення розмірності і виділення ознак, а потім для класифікації цілей.

Квадратичні дискримінантний аналіз (QDA). Використовується в машинному навчанні та статистичної класифікації для окремих вимірів двох або більше класів об'єктів або подій. Це більш загальний варіант лінійного класифікатора. QDA є параметричним підходом в контрольованому навчанні, яка моделює ймовірність кожного класу, як розподіл Гаусса.

Максимальний класифікатор ентропії (поліноміальний логістична регресія). У статистиці, максимальна модель ентропії класифікатора є моделлю регресії, яка узагальнює логістичну регресію, дозволяючи більше двох дискретних результатів. Це формує модель, яка використовується для прогнозування ймовірності різних можливих результатів категорично розподіленого залежно змінного, враховуючи безліч входів залежних змінних.

Метод K-найближчих сусідів. В області малюнка зізнання, к-найближчих сусідів алгоритму (к-NN) являє собою метод класифікації об'єктів на основі найближчих прикладів навчання в просторі. K-NN є тип, наприклад на основі навчання, де функції обчислення не відкладені до класифікації. Цей

алгоритм є одним з найпростіших алгоритмів машинного навчання, в якому об'єкт класифікується з використанням більшістю голосів своїх сусідів, і об'єкту потім присвоюється клас, який є найбільш поширеним серед його найближчих сусідів.

Наївний байесовский класифікатор. Даний класифікатор є простим, імовірнісним і статистичним класифікатором, який заснований на теоремі Байеса з сильними припущеннями (наївна. Як байєсовські класифікатори статистичного характеру, вони можуть передбачити ймовірність даного зразка, що належить до конкретного класу. Ймовірності прилеглої модель даного класифікатор можна назвати більш підходящою як «незалежної моделі ознаки», тому що наївний байесовский класифікатор передбачає, що вплив значення атрибута на даний клас не залежить від значень інших атрибутів. Таке припущення називається клас умовної незалежності. Це зроблено для спрощення обчислень, що беруть участь і, в цьому сенсі, вважається «наївним».

Метод опорних векторів. SVM - це відносно новий метод класифікації, що дозволяє уникнути двох класичних пасток машинного навчання: «обчислювальна здатність виживати, проектуючи дані в трильйонні вимірювання, і статистична здатність виживати, що на перший погляд виглядає як класична пастка для перевантаження» [18]. Вони були застосовані до величезної кількості проблем [19], включаючи розпізнавання почерку, розпізнавання об'єктів, розпізнавання осіб і, звичайно ж, категоризацію тексту [20].

У загальному випадку SVM є ще одним методом поділу гіперплоскостей в моделі векторного простору, але відрізняється тим, що він чутливий до ризику, так як передбачуваний вектор не просто розділяє гіперплоскость, але розділяє гіперплоскость з найбільшим потенційним межею похибки (тобто розділяє гіперплоскость може бути зміщена на найбільшу відстань до введення нової помилки класифікації). Більш загальне формулювання включає використання нелінійної функції ядра для визначення поділяють просторів, відмінних від гіперплоскостей.

У 1995 році Кортес і Вапник представили алгоритм. У найпростішому вигляді машина опорних векторів візуалізується в двовимірному просторі з набором даних, що складається з двох різних класів, тобто квадрата і кола. Ці квадрати і кола розділені гіперплощиною, і оскільки машина опорних векторів є двовимірною, гіперплощина представляється у вигляді одновимірної лінії. Оскільки існує нескінченна кількість гіперплоскостей, які можуть розділяти кола і квадрати, машина опорних векторів зацікавлена в гіперплоскості, яка забезпечує максимальний запас, також відомий як гіперплощина максимального краю. Це означає, що гіперплощина, яка розділяє два класи якнайдалі один від одного. На малюнку 1.1 показаний приклад максимальної межі гіперплощини.

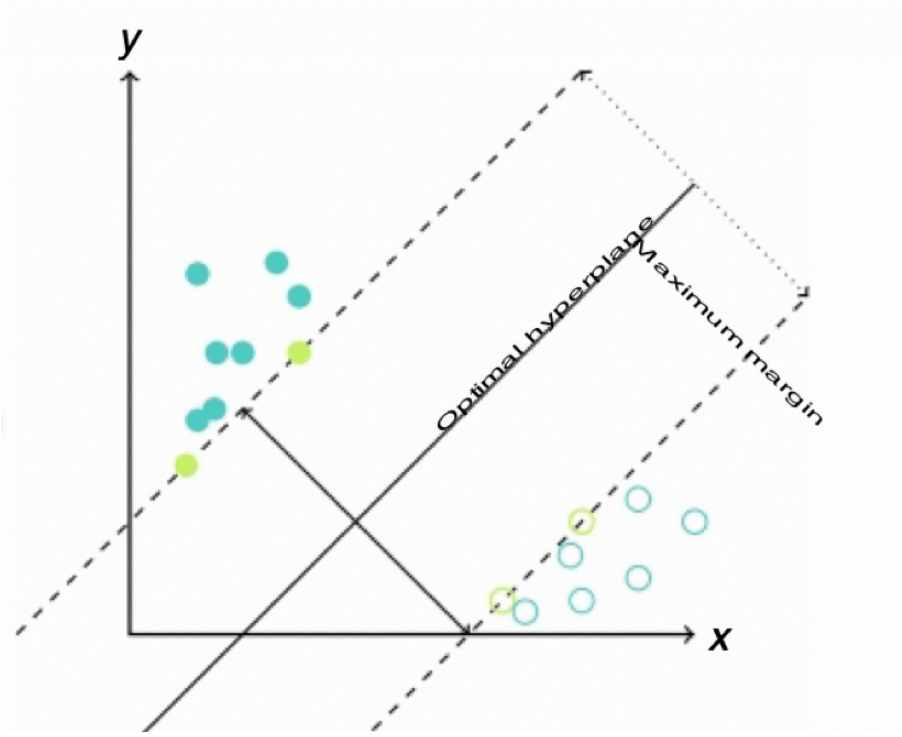


Рисунок 1.1 – Візуальне представлення методу опорних векторів

При навчанні класифікатора SVM побудови моделі будь-яка знайдена гіперплощина, яка розділяє два класи даних, може мати нульову помилку після завершення навчання. Однак модель з невеликою гіперплощиною, як правило, працює погано при класифікації даних з невидимого тестового набору. Тому моделі опорних векторів з великим запасом зазвичай мають кращі помилки

узагальнення, ніж моделі з меншим запасом. Є також випадки, коли точки даних не можуть бути розділені прямою лінією. Якщо це так, то застосовуються функії ядра. Функція ядра - це проекція точок даних в більш високий вимір, щоб мати можливість їх розділяти. Машини опорних векторів також популярні при ідентифікації авторів, оскільки алгоритм не позбавлений від прокляття проблеми розмірності.

Штучні нейронні мережі. Штучна нейронна мережа, скорочено ANN, являє собою сімейство навчальних алгоритмів, які в значній мірі натхненні своїм дизайном, щоб функціонувати так само, як людський мозок. Однак замість нейронів, аксонів, дендритів і синапсів ANN складається із сукупності залежних вузлів, які пов'язані один з одним в мережі. Найпростішим з відомих ANN є модель персептрона, в якій персептрон використовує два види вузлів: вхідні та вихідний. Вхідні вузли представляють вхідні атрибути і передають значення в вихідну посилання, вихідний вузол являє вихідні дані моделі і виконує всі обчислення. Як і людський мозок, кожен вузол в ANN відомий як нейрон, а в моделі персептрона ребра відомі як синаптическая зв'язок між вузлами. Кожному ребру між вхідним вузлом і вихідним вузлом присвоюється вага, це називається зваженим зв'язком і використовується для емуляції синаптичної сили між вузлами. На малюнку 1.2 показано, як може виглядати модель персептрона.

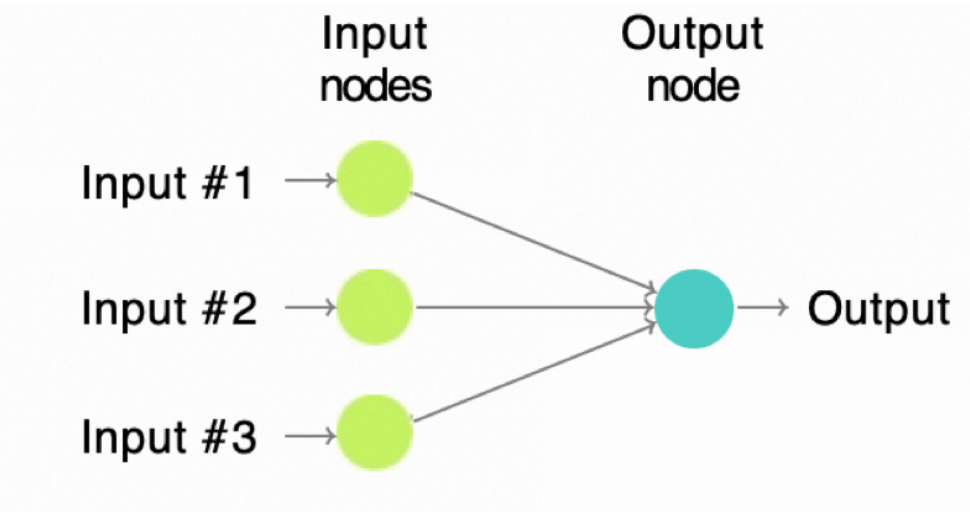


Рисунок 1.2 – Модель парсептрона

Щоб розрахувати вихідні дані моделі персептрона, для ваг і атрибутів виконується операція зваженої суми, а потім з суми віднімається коефіцієнт зміщення.

При навчанні моделі персептрона ваги по краях коригуються до тих пір, поки вихідні дані моделі персептрона не відповідатимуть фактичного виходу з навчальних даних. Недолік моделі персептрона полягає в тому, що вона працює тільки з лінійно-роздільними завданнями класифікації. Для задач класифікації, які не є лінійно нероздільними, повинна використовуватися більш складна модель. Ця модель відома як багатошаровий ANN і може мати кілька прихованих шарів між рівнем вхідного вузла і рівнем вихідного вузла. І замість того, щоб бути нейронною мережею з прямим зв'язком, де вузли в одному шарі пов'язані тільки з вузлами в наступному шарі, вона може бути періодичною, що означає, що вузли можуть бути пов'язані всередині шару або також пов'язані з попереднім рівнем, Як і в моделі персептрона, рівень вхідного вузла є атрибути, що підлягають оцінці, а рівень вихідного вузла представляє список доступних класів з навчальних даних. Прихований рівень вузла може, як рівня вихідного вузла, використовувати функцію активації, щоб знайти гіперплоскість і об'єднати її з вихідним вузлом, щоб мати можливість створювати вихідні значення, які є нелінійними. Схема, що показує нейронну мережу з прихованим шаром, показана на рисунку 1.3.

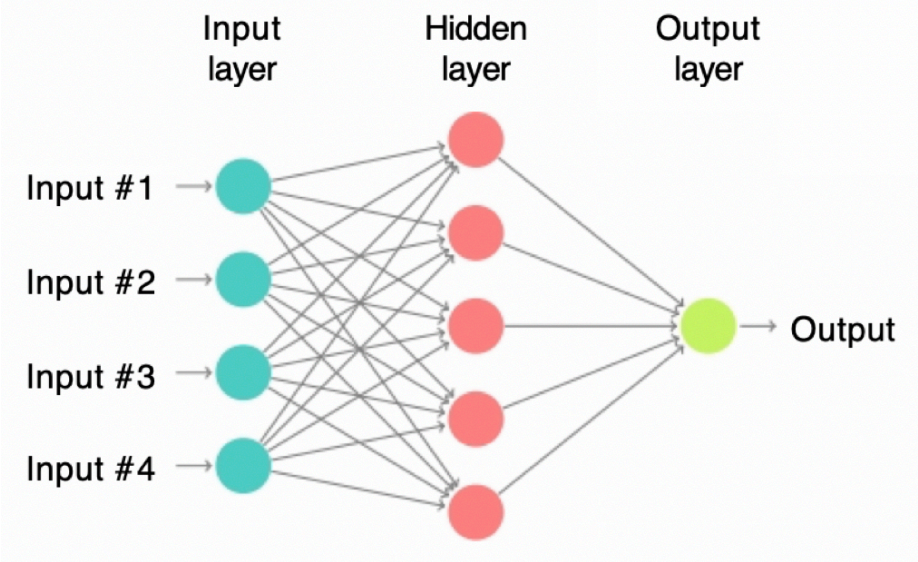


Рисунок 1.3 – Нейрона мережа з прихованим шаром

1.6 Висновок

У даному розділі розглянуто опис загальних теоретичних аспектів визнаяенна автора тексту. Були описані визначення, характеристики та властивості визначення авторського стилю, а також розглянуті основні підходи до їх розв’язання.

На основі аналізу існуючих методів визначення автора тексту було визначено, що методи засновані на машинному навчанні є більш ефективнішими в порівнянні з простою статистикою. Також було визначено, що вхідний текст потрібно піддавати певній обробці та представляти у вигляді, придатному до опрацювання математичними засобами. В ході досліджень наявних моделей машинного навчання було визначено, що найпопулярнішими є наступні: наївний баєсів класифікатор, метод опорних векторів, та штучні нейроні мережі для класифікації текстів.

2 РОЗРОБКА МАТЕМАТИЧНОЇ МОДЕЛІ

Даний розділ присвячений розробці математичної моделі визначення стилістичних характеристик при індентифікації автора даного тексту. Далі буде висвітлене дослідження та аналіз підходу для класифікації. Навдені висновки та запропоновані рішення базуються на результаті вивчення предметної області та аналізу існуючих засобів

В рамках дослідження стильометричного аналізу тексту, був використаний наступний алгоритм послідовних дій:

– вибір текстів для навчання;
– вибір текстових дескрипторів для аналізу - авторських відбитків;
– розрахунок характеристик для всіх дескрипторів, які використовуються для навчання нейронної мережі;
– специфікація мережі з її топологією і методом навчання;
– фактичне навчання мережі;
– тестування;
– аналіз отриманих результатів та підсумок висновків для поліпшення класифікації.

Описаний процес застосовувався кілька разів до різних вхідних даних, даний підхід був описаний в [22].

2.3 Вибір методу машинного навчання

Для класифікації текстів було обрано штучні нейронні мережі (ANN). З точки зору топології нейронні мережі можна розділити на дві категорії: прямі і рекурентні мережі. У мережах з прямим зв'язком, потік даних строго зв’язаний від вхідних до вихідних осередків, які можуть бути згруповані в шари, та не містить зворотного зв'язку. З іншого боку, рекурентні мережі містять петлі зворотного зв'язку, і їх динамічні властивості дуже важливі.

Найбільш популярним типом нейронних мереж, які використовуються в задачах класифікації, є мережа прямого зв'язку, яка побудована з шарів і має односпрямовані зважені зв'язки між нейронами. Типовими прикладами цієї категорії є мережі з багаторівневим персептроном або радіальною базисною функцією.

Тип багатошарового персептрона (MLP) більш точно визначається шляхом визначення кількості нейронів, з яких він побудований, і цей процес можна розділити на три частини, дві з яких описують кількість вхідних і вихідних нейронів, а третя частина специфікацію числа прихованих нейронів.

Кількість вхідних і вихідних нейронів може фактично розглядатися як зовнішня специфікація мережі, і ці параметри швидше знаходяться в специфікації завдання. З метою класифікації для аналізованих об'єктів визначено так багато відмінних ознак, що потрібно багато вхідних вузлів. Єдиний спосіб краще адаптувати мережу до проблеми - розглянути вибрані типи даних для кожної з обраних функцій. Наприклад, замість того, щоб використовувати абсолютне значення деякої ознаки для кожного зразка, оскільки це відносне значення має бути менше, ніж весь діапазон можливих значень.

Третій фактор в специфікації багатошарового персептрона - це кількість прихованих нейронів і шарів, і це важливо для здатності і точності класифікації. Без прихованого шару мережа здатна правильно вирішувати тільки лінійно роздільні завдання з вихідним нейроном, що розділяє вхідний простір на гіперплощину.

За допомогою одного прихованого шару мережа може класифікувати об'єкти у вхідному просторі, тоді як з двома прихованими шарами мережа може класифікувати будь-які об'єкти, оскільки вони завжди можуть бути представлені у вигляді суми або різниці деяких таких симплексів, класифікованих другим прихованим шаром.

Крім кількості шарів існує ще одна проблема з кількістю нейронів в цих шарах. Коли кількість нейронів невиправдано велика, мережа легко навчається, але погано узагальнює нові дані. З іншого боку, коли надто мало

прихованих нейронів, мережа може ніколи не дізнатися відносини між вхідними даними. Оскільки немає точного показника того, скільки нейронів слід використовувати при побудові мережі, звичайною практикою є побудова мережі з деяким початковим числом одиниць, і при поганому тренуванні це число або збільшується, або зменшується в міру необхідності.

Функція активації або передачі нейрона - це правило, яке визначає, як він реагує на дані, накопичені через його входи, які мають певну вагу. Зазвичай використовується лінійна або напівлінійна функція, жорстко обмежує порогова функція або плавно обмежує поріг, такий як сигмовидная або гіперболічна дотична. Через властиві їм ознаки (з яких найбільш важливими є вони лінійні, безперервні або диференціюються), різні функції активації виконуються з різною ефективністю в рішеннях для конкретних завдань.

Для задач класифікації сигмоид є найбільш часто використовуваною функцією активації, яка визначається за формулою:

$$y\,(n) = \frac{1}{1+e^{-\beta n}},$$

де n (нетто) - скалярний добуток, який фираховується за формулою:

$$n = W * X = W^{T}X = \sum_{J=0}^{J} W_j x_j \ ,$$

де ваги W і вхідних векторів X з j = 0, зарезервованим для зміщення t, шляхом установки x0 = 1 і w0 = - t.

2.2 Правила навчання

Щоб створити бажаний набір вихідних станів щоразу, коли набір вхідних даних представляється нейронної мережі, він повинен бути налаштований шляхом установки сили взаємозв'язків, і цей крок відповідає процедурі навчання мережі. Правила навчання грубо розділені на три категорії контрольованих, неконтрольованих і допоміжних методів навчання. Таким чином, в разі контрольованого навчання дані навчання вказуються у вигляді

пар вхідних значень і очікуваних вихідних даних. Шляхом порівняння очікуваних результатів з результатами, фактично отриманими з мережі, обчислюється функція помилки, і її мінімізація призводить до зміни ваг з'єднань таким чином, щоб отримати вихідні значення, які мають найтісніший контакт очікуваним для кожної навчальної вибірки і для всього навчального набору, при неконтрольованому навчанні відповідь не вказується, як очікується від нейронної мережі, і він залишає собі можливість виявити таку самоорганізацію, яка видає ті ж значення на вихідному нейроні для нових вибірок, що і для найближчої вибірки навчального набору.

Посилення навчання спирається на постійну взаємодію між мережею та її середовищем. Мережа не має ніяких ознак того, що від неї очікують, але вона може стимулювати її, виявляючи, які дії приносять найбільшу винагороду, навіть якщо це винагорода не є негайним, а затримується. Грунтуючись на цих нагородах, він здійснює таку реорганізацію, яка є найбільш вигідною в довгостроковій перспективі.

У широко використовуваних багатошарових мережах персептрон зазвичай застосовуються деякі варіанти методу контрольованого зворотного поширення. Класичний алгоритм зворотного поширення змінює вектор всіх ваг W відповідно до напряму спуску градієнта, який наведено в формулі:

$$\Delta W = -\eta \nabla e(W) ,$$

де η - швидкість навчання помилки, що виникає на виході мережі, наведеної в формулі:

$$e(W) = \frac{1}{2}\sum_{m=1}^{M}\sum_{i=1}^{I}(d_i^m - y_i^m(W))^2,$$

це сума помилок для всіх навчальних фактів M на всіх вихідних даних.

Нейрони, кожен з яких визначається різницею між очікуваним результатом тьмяності і тим, який генерується мережею (W).

Модифікація ваг, пов'язаних з мережевими з'єднаннями, може бути виконана або після кожної з навчальних вибірок, або після завершеної ітерації всього навчального набору.

Важливим фактором у цьому алгоритмі є швидкість навчання η, значення якої при занадто високому може викликати коливання навколо локальних мінімумів функції помилки, а при занадто низькому призводить до повільної збіжності. Ця місцевість вважається недоліком методу зворотного поширення, але його універсальність є перевагою.

Оцінка точності класифікації. Для більш точного визначення ефективності роботи алгоритму доцільніше буде мати декілька метрик, що відображатимуть якість віднесення висловлювань до однієї з двох категорій на тестовому наборі фраз:

– кількість позитивних результатів – TP;
– кількість негативних результатів – TN ;
– кількість помилкових позитивниз результатів – FP;
– кількість помилково негативних результатівх – FN.

Для основних метрик якості було запропоновано наступні:

– **accuracy** метрика. Частка коректно класифікованих документів від загального числа документів, що подаються на вхід класифікатору:

$$\text{Accuracy} = \frac{\text{TP+TN}}{\text{TP+TN+FP+FN}};$$

– **precision** (точність). Частка об’єктів, що класифіковані алгоритмом до конкретного класу, та які дійсно відносяться до нього:

$$\text{Precision} = \frac{\text{TP}}{\text{TP+FP}};$$

– **recall** (Повнота). Частка об’єктів конкретного класу, що вірно класифікована алгоритмом і віднесена до нього:

$$\text{Recall} = \frac{\text{TP}}{\text{TP+FN}};$$

– **f – measure** (F - міра). Середнє гармонічне для метрик Precision та Recall:

–

$$\text{F} = \frac{2*\text{Precision} * \text{Recall}}{\text{Precision+Recall}};$$

– статистика. **Капа Коена**. Величина до якої згода, що, спостерігається перевищує випадкову згоду і виглядає у вигляді пропорції максимального поліпшення (без впливу випадку), якого вдалося досягти двома або більше дослідниками при вимірюванні одного і того ж феномена:

–

$$\text{k} = \frac{\frac{\text{TP+TN}}{\text{TP+TN+FN+FP}} - \text{Pr (e)}}{1 - \text{Pr (e)}},$$

де відсоток випадкої згоди Pr(e) вираховується за формулою

$$\text{Pr(e)} = \text{P}_{\text{так}} + \text{P}_{\text{ні}},$$

де $\text{P}_{\text{так}}$ вираховується за формулою

$$\text{P}_{\text{так}} = \frac{\text{TP+FP}}{\text{TP+TN+FN+FP}} * \frac{\text{TP+FN}}{\text{TP+TN+FN+FP}},$$

а $\text{P}_{\text{ні}}$ вираховується за формулою

$$P_{\text{ні}} = \frac{FN+TN}{TP+TN+FN+FP} + \frac{FP+TN}{TP+TN+FN+FP}.$$

2.3 Використані тексти

У дослідженнях були використані тексти відомих українських письменників. Їх романи і короткі роботи дають досить широкий корпус текстів, щоб гарантувати, що характерні риси, виявлені на основі даних навчання, можуть дати узагальнені знання, що будуть використовуватися для підтвердження або виключення автора, що розглядається.

Очевидно, що літературні тексти можуть сильно відрізнятися по довжині, більш того, на всі стилістичні особливості можуть впливати не тільки різні часові рамки, в яких текст написаний, але і його жанр. Перша з цих проблем легко вирішується шляхом поділу довгих текстів, таких як романи, на кілька дрібніших частин приблизно однакового розміру.

Даний підхід дає додаткову перевагу в задачах класифікації, так як навіть у разі деяких неправильних результатів класифікації цих частин весь текст все ще може бути належним чином приписаний автору, присвоюючи остаточне рішення на більшості результатів.

Для передбачуваної реалізації класифікатора з штучними нейронними мережами, які ефективно працюють з великим обсягом даних, додавання вибірок в навчальний набір просто означає кращий охоплення простору введення, що важливо в безперервному випадку.

У таблиці 2.1 представлена коротка характеристика використовуваних текстів для кожного автора. Більш детальній опис використаних текстів для тренування та тестування описаний в Додатку Б.

Таблиця 2.1 – Коротка характеристика використаних текстів для тренування та тестування

№	Автор	Кількість слів	Кількість знаків пунктуації
1	Іван Франко	28737	3561
2	Остап Вишня	29674	3965
3	Володимир Виниченко	31675	4001
4	Марко Вовчок	30203	4123
5	Павло загребельний	29978	3989
6	Григір Тютюник	32115	4314
7	Микола Хвильовий	30385	4678
8	Іван Нечуй-Левицький	24268	4251
9	Анатолій Дімаров	37893	3867
10	Леся Українка	38564	5154
11	Михайло Коцюбинський	29726	4941
12	Олександр Довженко	30386	4618
13	Ліна Костенко	28421	4257
14	Юрій андрухович	31893	4732
15	Іван Драч	32649	3368
16	Валерій Шевчук	28754	4567
17	Тарас Шевченко	37493	4876
18	Валер’ян Підмогильний	29968	4475
19	Олег Гончар	31291	4159
20	Костецький Анатолій	30589	4954

2.4 Вибір вхідних даних

Однією з основних складностей в рішенні задач класифікації, що базуються на методах навчання по прецедентах являється пошук необхідних даних, на яких ми зможемо навчати нашу систему.

Встановлення особливостей, які працюють як ефективні дискримінатори досліджуваних текстів, є однією з найважливіших проблем в дослідженні авторського аналізу.

У роботі використовуються, лексичні і синтаксичні текстові дескриптори використання функціональних слів, та використання розділових знаків. Список функціональних слів був обраний після аналізу дослідження [43]. Причиною використання їх списку функціональних слів було те, що вони заявили, що немає хороших правил для вирішення проблеми, які функціональні слова включати. Таким чином за основу був обраний список функціональних слів, який вони використовували для своїх досліджень, попередньо перекладений та адаптований під українську мову. Обрані дескриптори описані в таблиці 2.2.

Таблиця 2.2 – Обрані характеристики тексту

Найменування	Обрані дескриптори
Функціональні. слова	про, вище, згодом, все, хоча, є, серед, і, інше, будь-хто, будь-що, як, бути, тому що, перед, позаду, внизу, поруч, між, обома, але, за, може, вниз, кожен, або, достатньо, все, мало, слідуючи, за, з, має, він, її, його, якщо, в тому числі, всередині, в, це, його, останнє, менше, мало, багато, я, більше, більшість, повинно, моє, поруч, не потрібно, ні, ніхто, не, нічого, з, на, раз, один, на, протилежний, або, наш, зовні, над, власний, минуле, за, багато, плюс, щодо, те саме, кілька, вона, повинна, оскільки, так, хтось, щось, таке, ніж, що, то, їх, це, вони, це, ті, хоча, через, до, назустріч, під, якщо, на відміну від, поки, до, нас, використовується, через, ми, що, коли, де, чи, який, поки, хто, чий, буде, з, в межах, без, варто, так, ти, твій
Знаки пунктуації	”,”, ”.”, ”?”, ”!”, ”:”, ”;”, ”’”, ””””

2.5 Архітектура нейронної мережі

У якості структури нейронної мережі було обрано мережу зворотного поширення, яка зображена на рисунку 2.1. Кількість вхідних нейронів дорівнює кількості обраних характеристик. Приховані шари зв'язані прямим зв'язком з сигмоїдальною функцією активації. Нейрона мережа містить два приховані шари, а мультиплікатор швидкості навчання був не більше 0,4 для всіх розпізнаних зразків.

В архітектурі нейронної мережі було використано два виходи. Насправді, було б можливо використовувати один вихід і шляхом інтерпретації його активного стану як одного класу і неактивного стану виходу як другий клас. При такій архітектурі завдання також було б вирішене, але при цьому підході текст завжди був би приписаний або одному, або іншому автору, а бінарна класифікація з невирішеним вердиктом неможлива.

Два виходи дозволяють розпізнати ситуацію, коли мережа не може легко розпізнати стиль письма будь-якого з раніше навчених авторів і не може правильно класифікувати деякі зразки тексту.

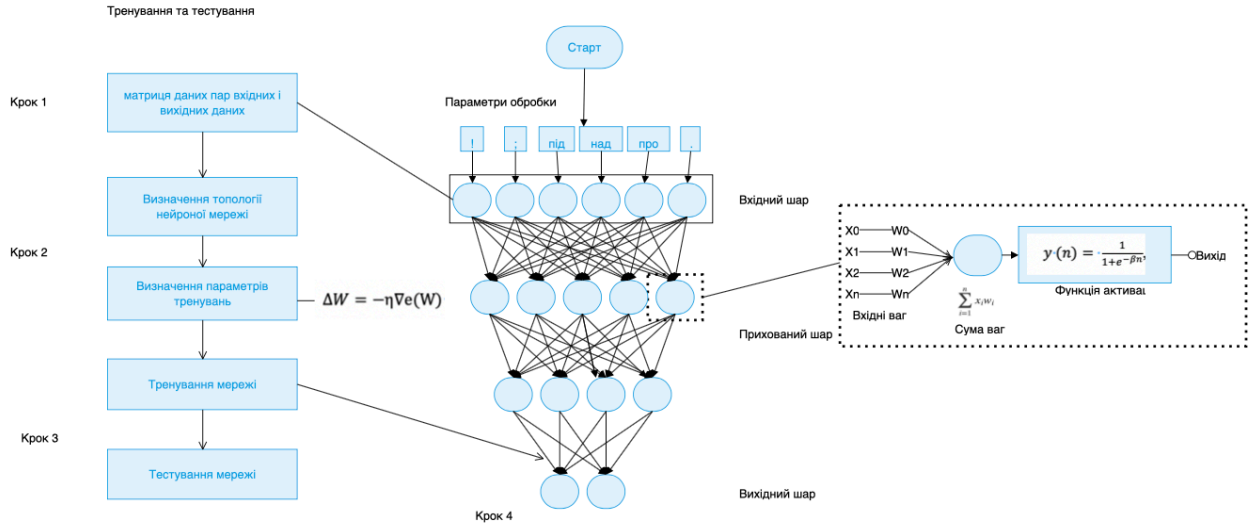


Рисунок 2.1 – Архітектура класифікатору

2.6 Висновки

В даному розділі була описана та розроблена методика визначення автора тексту на основі синтаксичних та лексичних текстових ознак, які будуть використовуватися.

В якості метода класифікації для подальшого дослідження було обрано штучні нейроні мережі. Описано методологію навчання текстового класифікатора, пошуку оптимальної кількості структурних одиниць, зокрема кількості прихованих шарів та виходів нейроної мережі. Навчання та параметрів регуляризації. Розглянуто способи оцінки побудованої моделі та представлена загальна архітектура нейроної мережі.

Було складено 2 комплектів характеристик, згідно яких можна добути ознаки авторського стилю з тексту. Основне питання подальшого дослідження – визначення кращого з складених комплектів. Заздалегідь визначити найоптимальніший варіант без практичних випробувань доволі складно. Для підтвердження висунутих пропозицій їх необхідно перевірити на практиці.

3 РОЗРОБКА ПРОГРАМИ НА ОСНОВІ ОБРАНОЇ МОДЕЛІ

У цьому розділі показано, як структура ідентифікації авторства, представлена в розділі 2, використовувалася в архітектурі реалізованої системи, включаючи збір даних, витяг ознак і генерацію моделі.

3.1 Аліз вимог до підсистеми

Для якісного та своєчасного впровадження системи завжди важливо пояснити архітектуру системи. Опис архітектури має важливе значення для вдалого планування ресурсів, планування самого процесу розробки, необхідного в процесі експлуатації та документації. Необхідно мати опис архітектури системи для подальшого процесу розробки, розширення функціональності, модифікації або модернізації системи.

Сьогодні існує багато вимог до програмного забезпечення та інформаційних систем. Вимоги до швидкості розробки також вимагають дуже високого темпу, часто вимагаючи впровадження нових функціональних можливостей. Програмний продукт повинен бути готовими до швидкого масштабування. Усе це в поєднанні з високим ступенем складності сучасних виробничих систем вимагає вдосконалення процесів розробки та самої структури та архітектури системи вцілому. Для розробки програмного продукту, потрібно з самого початку проаналізувати вимоги, яким повина відповідати розроблювальна система.

Розроблювальна система повина відповідати вимогам поставленим в даній дипломній роботі, а саме на вхід подається певна кількість текстів для кожного автора та текст, який буде індентифікуватися. На виході, згідно прийнятих навчальних даних, система повина видати результат припущення приналежності тесту запропонованим авторам.

Для більш успішнігошо формулювання вимог, їх потрібно розділити на функціональні та нефункціональні вимоги. До функціональних вимог буде віднесено те, що стосується алгоримів роботи системи та її поведінки. До нефункціональних вимог буде віднесено те, що стосується характеру

поведінки системи. Робота система повина відповідати висунитим гіпотезам, та виходячи з цього до функціональних вимог можна віднести:

- можливість зчитування файлу в форматі txt, docx, pdf;
- можливість відображати час, затрачений на певний етап роботи;
- можливість відображати результат класифікації у вигляді заповненого файлу;
- можливість збереження розпізнаного тексту до текстового файлу;
- можливість подальшої міграції в інші додатки у вигляді окремого модулю.

До нефункціональні вимоги віднесено:

- можливість використання на комп’ютерах з операційною системою Mac OS та Microsoft Windows;
- програма повинна мати інтуїтивний та зручний інтерфейс для внесення даних та аналізу результатів.

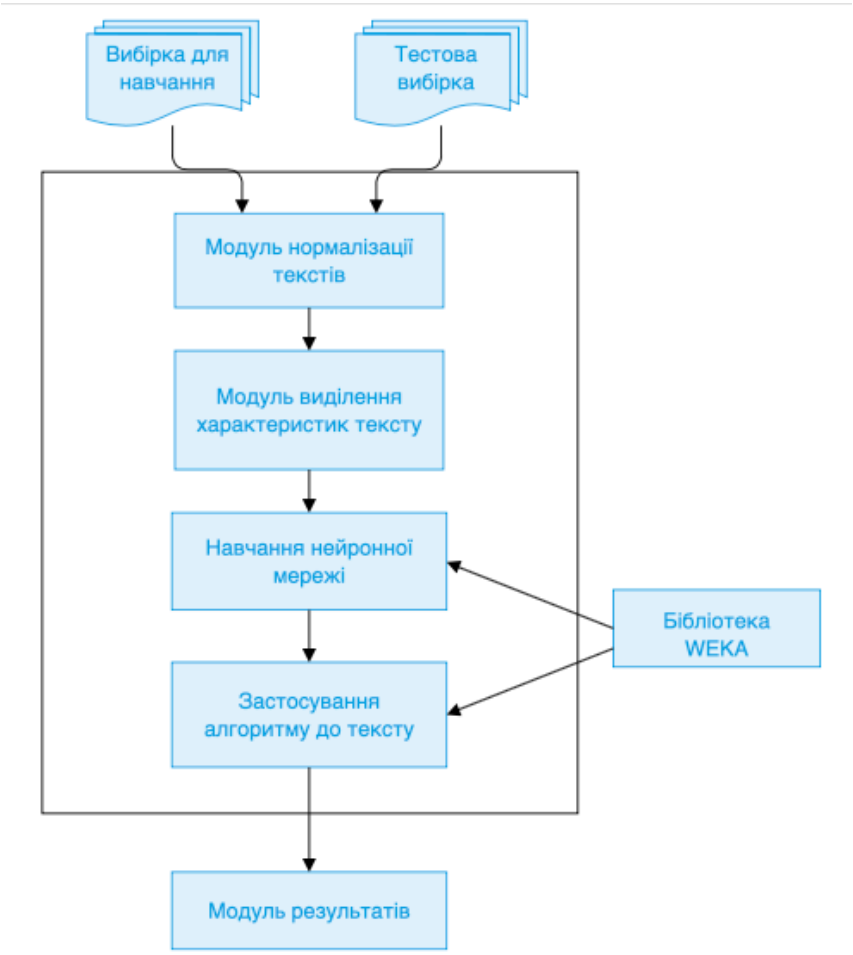
3.2 Архітектура системи

Для того, щоб продемонструвати функції запропонованого підходу та підтвердити гіпотезу, була розроблена програма, здатна класифікувати та виділяти потрібні дескриптори з текстів. Поданий тестовийтекст після обробки, буде класифікований та віднесений до певного автора. На рисунку 3.1 схематично зображено загальну роботу програми, яку можна предсавити у вигляді послідовності кроків.

У якості мови програмування було обрано Java, яка є відносно популярною для реалізації програм, які використовують машине навчання. Використання даної мови дало можливість використовувати безкоштовні бібліотеки, значно скорочуючи цим час на реалізацію алгоритмів.

Основними компонентами програми є модуль, який визначає та фіксує описані раніше властивості тексту, і модуль, який використовує описаний метод для навчання та подальшої класифікації.

Цілью програми є показати дієздатність запропонованого підходу визначення автора тексту, тому вона зроблена достатньо просто. На рисунку 3.1 зображена схема роботи програми.



Рисунко 3.1 – Схема роботи програмного забезпечення

Після того, як дані занесені до програми, вони піддаються попередній обробці. Представлений набір даних потрібно нормалізувати перед проведенням подальшого аналізу. Попередня обробка тексту необхідна для

використання чистіших даних для навчання без зайвих символів або пробілів, які будуть впливати на тренування нейронної. Весь текст переводиться до нижнього регістру та за допомогої регулярних виразів видаляються специфічні символи.

Після видалення зайвих пробілів та знаків текст проходить етап нормалізації, відомий як стемінг. Для даного етапу був використаній стемінг Портера. Для нього був використаний клас PorterStemmer. На рисунку 3.3 показано фрагмент коду для стемінгу вхідних текстів

```
public Set<String> GetStemmerText(String text) {
    // отримати слова
    String[] words = clean(text).split( regex: "[ \n\t\r$+<>№=]");

    // видалити закінчення
    for (int i = 0; i < words.length; i++) {
        words[i] = PorterStemmer.doStem(words[i]);
    }
```

Рисунок 3.2 – Фрагмент коду для стемінгу текстів

Для класифікатора потрібна нам інформація ґрунтується на стилі написання. Для кожного тексту отримується певні інформацію, яка буде ознаками класифікатора. Для цієї частини використовується лише стандартні бібліотеки Java, щоб маніпулювати файлами, рядками та символами. Робота збору даних виконується за допомогою двох основних класів: DataRepresentation та TextData.

DataRepresentation використовується для розбору та підрахунку функцій в текстах. Для перевірки використовується хеш-карта об'єктів і аналіз функції, та обчислює синтаксичний аналіз і повертає об'єкти і їх кількість в об'єкті TextData.

TextData - це об'єкт, який містить всю інформацію про певний документ, та дозволяє класифікатору швидко отримати дані. На рисунку 3.3 показано метод отримання характеристик тексту.

```
public static TextData parse(Article article) {
    byte[] bytes = null;
    int paragraphs = 1;
    HashMap<Character, Integer> punctuation = new HashMap<Character, Integer>()
    { {';', 0}, {'!', 0}, {'?', 0}, {':', 0}, {',', 0}, {'.', 0}, {'"', 0}, {'–', 0} };

    try {
        bytes = Files.readAllBytes(Paths.get(article.getCurrent().getPath()));
    } catch (IOException e1) {
        e1.printStackTrace();
    }
    String text = new String(bytes, StandardCharsets.UTF_8);
    for(int i = 0; i < text.length(); i++) {
        Character c = text.charAt(i);
        if(punctuation.containsKey(c)) {
            punctuation.put(text.charAt(i), punctuation.get(c) + 1);
        }
    }

    String[] lines = text.split("\n");
    ArrayList<Integer> numberOfWords = new ArrayList<Integer>();
    for (String line : lines) {
        numberOfWords.add(line.split(" ").length);

    }
    return new TextData(article.getAuthor(), punctuation, numberOfWords,  text.length(), lines.length);
}
```

Рисунок 3.3 – Метод отримання характеристик тексту

Для реалізації класифікатора використовувалася бібліотека Java: Weka.

Weka - це бібліотека машинного навчання, розроблена в університеті Вайкато, Нова Зеландія, та являє собою найбільш відому бібліотеку Java. Це універсальна бібліотека, яка здатна вирішувати широкий спектр завдань машинного навчання, таких як класифікація, регресія і кластеризація.

Всього в реалізації було використано 84 лексичні та синтаксичні ознаки, вони представлені у вигляді числових значень у форматі, відомому як ARFF.

ARFF-файл, відомий, як формат файлу відношення атрибутів, який складається з поля заголовка та поля даних. У полі заголовка присутні назва відношення та перелік усіх використовуваних функцій, а в полях даних, який має бути класифікований кожен екземпляр, представлений як рядок значень. Ці значення можуть бути комбінацією рядків, дат. Приклад ARFF-файлу можна побачити у Додатку В

У процесі реалізації кожен текст обробляється індивідуально через об'єкт вилучення об'єктів, який описано вище, а вилучені функції записуються в ARFF-файл, як екземпляри. На рисунку 3.4 шлях обробки тексту.

Рисунок 3.4 – Процес обробки тексту

Припущення про приналежність тестових текстів записується до призначеного для цього файлу у вигляді певного числового значення, що лежить в межах від 0 до 1, при чому чим значення ближче до 1, тим більша ймовірність, що відповідний текст належить певному автору, і навпаки, якщо значення наближається до 0.

3.3 Висновки

У даному розділі, що був присвячений розробці прошрамного забезпечення з обраним в розділі 2 набором функцій. Було складено та описано функціональні та нефункціональні вимоги до програмного забезпечення. Також описано базову схему роботи програмного забезпечення. Мовою реалізації обрано мову Java, а для реалізації необхідних функцій було використано готові рішення у вигляді безкоштовних бібліотек. Отримане застосовання може бути використане як окремий модуль для інтеграції в інші системи з мінімальними змінами. Наступний розділ буде присвячений дослідження впливу різних параметрів на результат. В ході дослідження будуть випробувані різні моделі, з яких буде обрана найкраща.

4 РЕЗУЛЬТАТИ РОБОТИ МОДЕЛЕЙ ТА ТЕСТУВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

У цьому розділі ми розглянемо, проведені експерименти та їх відповідні результати.

Для тестування розробленої методології було проведно сценарій, який розглядає набір даних з обмеженим числом авторів, де кожен автор представлений з достатньою кількістю навчальних і тестових даних.

Використовуваний набір даних складається з 20 різних авторів з максимум 20 творами для кожного та з змішаним нобором синтаксичних та лексичних текстових ознак, які наведені в теблиці 2.2.

Перевірка наборів даних проходила з різною кількістю статей на автора, від п'яти до двадцяти. Причиною для цього було перевірити те, як класифікація по кожному набору виконується з різним розміром даних для навчання та тестування.

Для даних були виконані чотири ітерації класифікаторів, перша ітерація використовувала тільки п'ять наукових робіт на автора в якості навчальних і тестових даних, в той час як друга ітерація використовувала десять дослідних робіт, третя ітерація використовувала 15 творів на автора, а четверта ітерація використовувала 20 текстів для кожного автора.

Для проведеного сценарію, було використано три різні метрики, про які було описано в розділі 2, а саме: показник успішності, який являє собою число примірників, які класифіковані правильно. Другою метрикою, яку ми використовували, була F -міра, та остання метрика, яка була розглянута - це Каппа Коена статистика, яка вимірює внутрішню згоду класифікатора. Для зручності представлення даних в графічному вигляді результати класифікацій по кожній метриці були представлені в відсотках.

Результат точності класифікації з 5 творами для кожного автора наведені на рисунку 4.1. Загальна кількість тренувальних текстів становить 100.

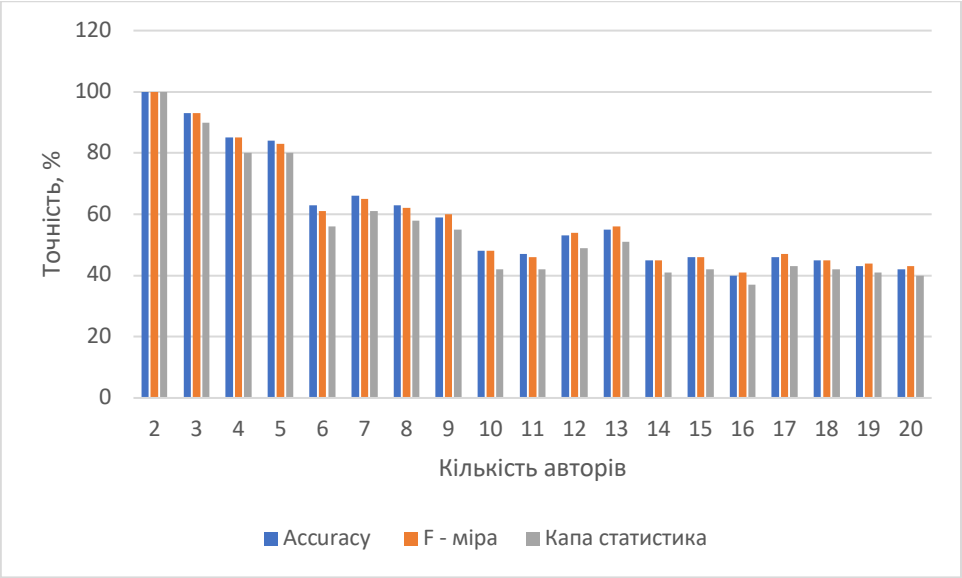


Рисунок 4.1 – Точність класифікації для 5 робіт на автора

Результат точності класифікації з 10 творами для кожного автора наведені на рисунку 4.2. Загальна кількість тренувальних текстів становить 200.

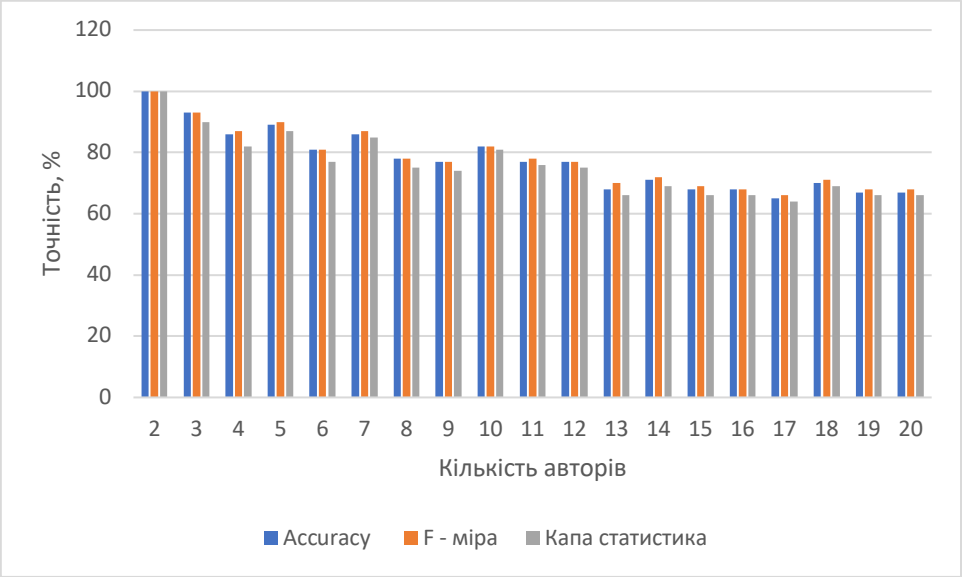


Рисунок 4.2 – Точність класифікації для 10 робіт на автора

Результат точності класифікації з 15 творами для кожного автора наведені на рисунку 4.3. Загальна кількість тренувальних текстів становить 300.

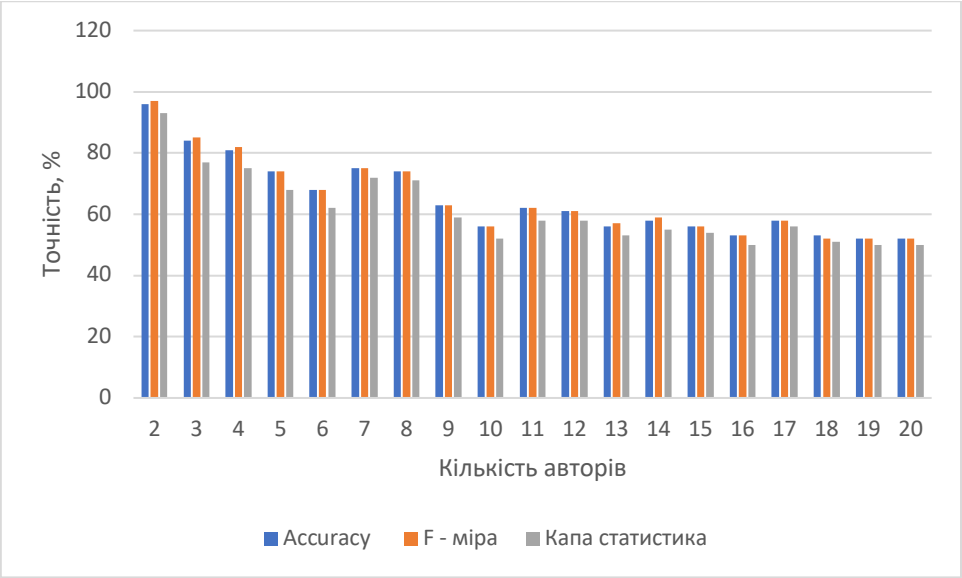


Рисунок 4.3 – Точність класифікації для 15 робіт на автора

Результат точності класифікації з 20 творами для кожного автора наведені на рисунку 4.4. Загальна кількість тренувальних текстів становить 400.

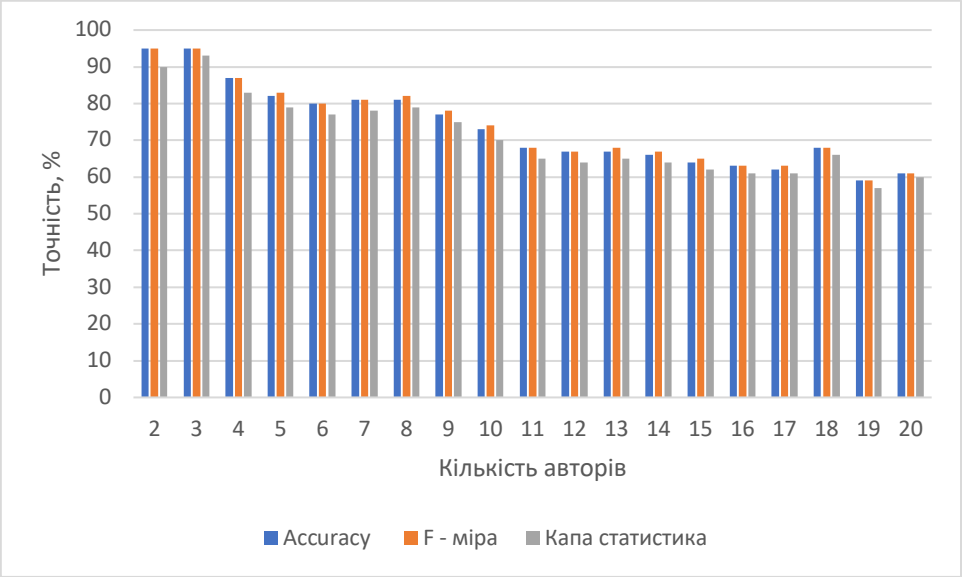


Рисунок 4.4 – Точність класифікації для 20 робіт на автора

З результатів, наведених на рисунках 4.1 – 4.4, спостерігається, що алгоритм штучної нейронної мережею з багатошаровим персептроном показав себе добре, проте кожен раз, коли ми додавали нового автора, точність класифікації падала.

Для завдання з двома авторами, де розглядається віднесення твору одному з двох авторів, класифікатор досягає максимальної точності 97% з показником Каппа і показником F1, рівним 1. Як видно з наведених вище графіків результатів для ситуації з двадцяти авторів досягається максимальна точності 67% з показником каппа 0,66 і показником F1 0,68. Основна ідея використання набору з двадцятьма авторами, полягалає в тому, що це була найважча проблема для алгоритмів класифікації, і, отже, вона дає найбільш важливий результат.

Проведені результати показують, що збільшення кількості навчальних даних для класифікатора робить клас більш ефективним.

Для першого набору даних ми почали тестування з п'ятьма дослідницькими роботами на автора для нашої десятиразової перехресної перевірки і збільшили розмір набору даних до десяти дослідницьких робіт на автора. При розгляді всіх двадцяти авторів в наборі даних ми побачили явне збільшення продуктивності, яке можна прослідити з графіків.

Що стосується точності, то кількість правильно класифікованих текстів покращився на 61%, для F-міра - на 64%, а для показника Каппа - на 72%. В цілому це підтверджує те, що продуктивність класифікаторів збільшується, коли збільшується кількість навчальних і тестових випадків.

4.1 Дослідження впливу додаткової обробки тексту на якість класифікації

Так як точність показників роботи модуля на даному етапі не досить висока, як для такого роду систем було вирішено знайти додаткові рішення, що потенційно зможуть покращити результати розпізнавання.

Як вже було сказано напочатку даного розділу в підрозділі про результат дослідження предметної області, вхідний текст піддавався лише базовій обробці, яка включала в себе переведення усіх літер до нижнього регістру.

Перше, що повинно гарно вплинути на результат, це використання стемінгу в обробці текстів. Стемінг - це процес скорочення слова до основи

шляхом відкидання допоміжних частин, таких як закінчення чи суфікс. Результати стемінгу іноді дуже схожі на визначення кореня слова, але його алгоритми базуються на інших принципах. Тому слово після обробки алгоритмом стемінгу (стематизації) може відрізнятися від морфологічного кореня слова. Це допоможе відкинути зайві символи, з яких власне і будуть утворюватися n-грами, а такі символи практично не несуть ніякого змістового навантаження.

Наступним кроком було вирішено видалити зайві службові символи та випадкові символи, так як вхідний набір текстів має певні хеш теги, помарки і тд. Regex (регулярні вирази) підхід було використано для виконання розбору фраз та викорінення спеціальних символів.

Після впровадження запропонованих варіантів, що повинні потенційно покращити результат були зроблені нові заміри згідно обраних показників для оцінки якості моделі. На риснунку 4.5 наведені результати класифікації текстів після додаткової обробки.

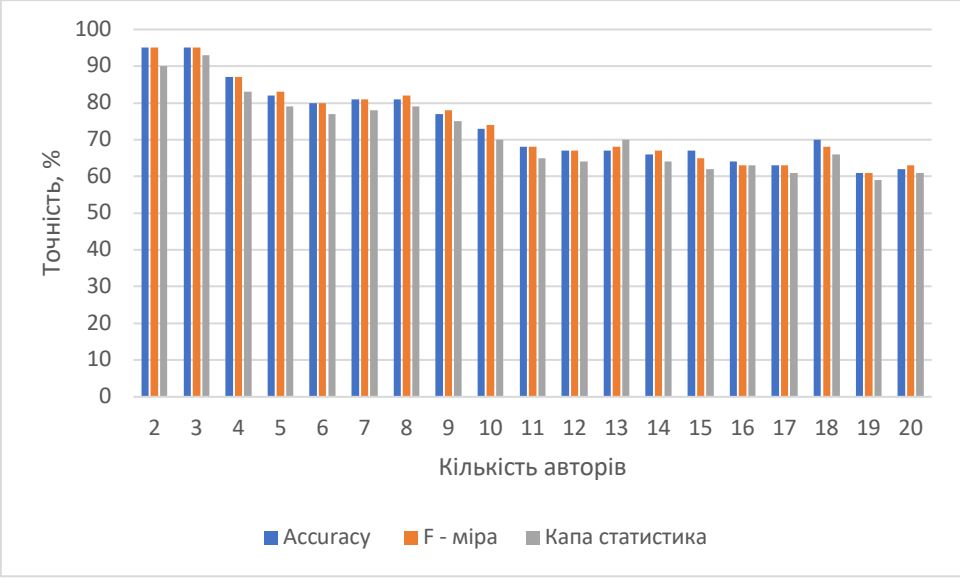


Рисунок 4.5 – Результат класифікації після додаткової обробки текстів

4.2 Вплив розроблених моделей на результат класифікації

Для вияву впливу обраних дескрипторів на результат виявлення автора тексту, був проведений експримент, який включав в себе послідовне додавання на вхід нейроної мережі нових дескрипторів. Щоб дослідити вплив дескрипторів на точність класифікації, вони були поділені на такі набори:

– набір 1, який далі буде позначатися, як N1 та буде містить набір з чотирьох знаків пунктуації: ”,”, ”.”, ”?”, ”!”;

– набір 2, який далі буде позначатися, як N2 та буде містить набір з інших чотирьох знаків пунктуації: ”:”, ”;”, ”’”, ”” ”;

– набір 3, який далі буде позначатися, як N3 та буде містить набір з наборів N1 та N2: ”,”, ”.”, ”?”, ”!”, ”:”, ”;”, ”’”, ”” ”;

– набір 4, який далі буде позначатися, як N4 та буде містить набір з функціональних слів, які описані в таблиці 4.2.

Результат точності класифікації для кожного з наборів дескрипторів наведений в таблиці 4.1. Як було показано на початку розділу, чим більша кількість тренувальних творів використовуеться, тим вища точність класифікації, тому для даного експерименту було використано максимальна кількість творів на кожного автора, а саме 20 творів, та визначення автора з поміж 20 авторів. Загалом для тренування було використано 400 творів. В якості метрики результатів класифікації було використано accuracy метрику.

Таблиця 4.1 – Результати класифікації текстів з різними наборами вхідних моделей

Автор	Точність класифікації для N1, %	Точність класифікації для N2, %	Точність класифікації для N3, %	Точність класифікації для N4, %
Іван Франко	59,1	55,2	61	58,3
Остап Вишня	58,6	57	63,4	60,2
Володимир Виниченко	53,4	49,9	56,5	57
Марко Вовчок	61,8	53,1	64,9	62,6
Павло Загребельний	61,2	56,2	64,1	63,1
Григір Тютюник	59,1	51,2	60,8	61
Микола Хвильовий	58,2	55,4	60,8	59,5
Іван Нечуй-Левицький	65,4	55,3	66,2	63,5
Анатолій Дімаров	61,2	60	64,2	62,7
Леся Українка	66,8	62,1	71,2	64,8
Михайло Коцюбинський	62,1	58,9	63,7	57,4
Олександр Довженко	61,6	58,1	67,3	60,3
Ліна Костенко	64,1	60	66	59,7
Юрій андрухович	63,1	60,1	68,8	69,1
Іван Драч	65,6	59,2	68,7	62,2

продовження таблиці 4.1

Автор	Точність класифікації для N1, %	Точність класифікації для N2, %	Точність класифікації для N3, %	Точність класифікації для N4, %
Валерій Шевчук	57,1	53,1	59,9	66
Тарас Шевченко	66,7	64,1	68,9	56,2
Валер'ян Підмогильний	55,9	53,3	57,3	64,4
Олег Гончар	55,5	54	60,3	62,3
Костецький Анатолій	59,1	57,6	64,3	65,2

Провівши результати наборів з таблиці 4.1, можна зробити висновки, що найвищий коефіцієнт класифікації забезпечується використанням синтаксичних текстових ознак, а найменший коефіцієнт класифікації забезпечується використанням лексичних текстових ознак. Стиль написання може бути настільки специфічним і відмітним, що вимагає використання менш типових дескрипторів, наприклад, різних функціональних слів.

4.3 Висновки

В даному розділі було досліджено вплив параметрів моделі на результати класифікації. Проведені результати показують, що збільшення кількості навчальних даних для класифікатора робить клас більш ефективним.

Для першого набору даних ми почали тестування з п'ятьма дослідницькими роботами на автора для нашої десятиразової перехресної перевірки і збільшили розмір набору даних до десяти дослідницьких робіт на автора. При розгляді всіх двадцяти авторів в наборі даних ми побачили явне збільшення продуктивності, яке можна прослідити з графіків.

Було показано те, що найвищий коефіцієнт класифікації забезпечується використанням синтаксичних текстових ознак, а найменший коефіцієнт класифікації забезпечується використанням лексичних текстових ознак.

Було показано, що деякі роботи неправильно класифіковані, оскільки текстові функції, що описують їх, недостатньо точні для виконання завдання. Стиль написання може бути настільки специфічним і відмітним, що вимагає використання менш типових дескрипторів, наприклад, інших функціональних слів або використання інших тренувальних текстів.

Аналіз результатів дає розуміння того, використнаий підхід класифікації текстів дає результати в діапазоні від 55% до 90% при наявності великої кількості слів. Проте при класифікації текстів з малою кількістю слів корисної інформації в них значно менше, а значить і кількість ознак, що можна добути з них, також мала. Це в свою чергу негативно впливає на кінцевий результат.

5 РОЗРОБКА СТАРТАП ПРОЕКТУ

Ідея проекту полягає у створенні програмного продукту, який матиме механізм атрибуції тексту на основі стильометричних особливостей авторів та за допомогою нейронної мережі буде визначати автора ще невідомого програмі тексту. Така система стане у пригоді філологам, які проводять роботи з визначення авторів невідомих текстів. Автоматизована система дасть користувачам можливість проводити аналіз невідомих текстів та дозволить зменшити витрати часу.

5.1 Опис ідеї стартап-проекту

Проаналізуємо зміст ідеї, її можливі напрямки застосування, чим запропонована ідея відрізняється від існуючих аналогів, а також основні вигоди, які може отримати користувач товару. Результати аналізу представлені у таблиці 6.1.

Таблиця 5.1 – Опис ідеї стартап-проекту

Зміст ідеї	Напрямки застосування	Вигоди для користувача
Створити програмний продукт, який матиме механізм атрибуції тексту на основі стильометричних особливостей авторів та за допомогою нейронної мережі буде визначати автора ще невідомого програмі тексту	1. В роботі філологів	Зменшення витрат часу на встановлення відповідностей
	2. В учбових цілях	Можливість наглядно побачити стилістичні особливості кожного з авторів

На ринку існують аналоги подібних систем, але більшість з них розробляються лише для вирішення відповідної конкретної задачі, мають однонаправлену конвертацію. Ці аналоги в основному англомовні, дорогі або застарілі на сьогоднішній день, а деякі з них без «рідного» АПК не працюють окремо. До того ж розроблена автоматизована програмна система створена для цільової аудиторії вітчизняного ринку.

Тому доцільно проводити аналіз потенційних техніко-економічних переваг ідеї порівняно з пропозиціями конкурентів. Результат аналізу у таблиці 5.2.

Таблиця 5.2. – Визначення характеристик ідеї проекту

Техніко-економічні характеристики ідеї	Продукція конкурентів			Слабкі (W), нейтральні (N) та сильні (S) сторони		
Назва продукту	SqlTextines	WhiteAutorTown	MnMTK		✓	
Операційна система та версії	Крос-платформна	Платформи Windows	Платформи Windows		✓	
Системні вимоги	Мінімальні	Від 1 Гб ОЗУ	Від 2 ГБ ОЗУ		✓	
Розміри	-	380 Мб	400 Мб			✓
Мови програмування	Java	C++	C++			✓
Необхідність встановлення додаткового ПЗ	браузер	наявність АПК	наявність АПК			✓

Автоматизована система вже розроблена та представлена у вигляді програмного додатку у фоматі *.EXE. Розроблена система зовсім малого розміру (всього 10 Мб), потребує встановлення на комп’ютер для роботи, а також не потребує підключення до Інтернету.

Перевагами даної розробки є те, що більшість подібних програмних застосунків в світі створені для виключно комерційних цілей, ціни на такі програмні засоби зависокі.

5.2 Технологічний аудит ідеї проекту

Для проведення технічного аудиту ідеї проекту, потрібно провести аудит технологій, за допомогою яких можна реалізувати ідею проекту. І для початку потрібно визначити можливість технологічної здійсненності проекту. Результат представлений у таблиці 5.3.

Таблиця 5.3 – Технологічна здійсненність ідеї проекту

Ідея проекту	Технології її реалізації	Наявність технологій	Доступність технологій
Створити програмний продукт, який матиме механізм атрибуції тексту на основі стильометричних особливостей авторів та за допомогою нейронної мережі буде визначати автора ще невідомого програмі тексту	Середовище розробки файлів-застосунків Idea	✓	Доступно (платне)

Обрана технологія реалізації ідеї проекту: Microsoft Visual Studio 2019 – середовище розробки файлів-застосунків.

Обрана технологія доступна, не потребує доробки, а також безкоштовна та надає усі необхідні можливості для реалізації поставленої задачі.

5.3 Аналіз ринкових можливостей запуску стартап-проекту

Визначення ринкових можливостей, які можна використати під час ринкового впровадження проекту, та ринкових загроз, які можуть перешкодити реалізації проекту, дозволяє спланувати напрями розвитку проекту із урахуванням стану ринкового середовища, потреб потенційних клієнтів та пропозицій проектів-конкурентів. Для цього спочатку проводиться аналіз попиту (таблиця 5.4).

Таблиця 5.4 – Попередня характеристика потенційного ринку стартап-проекту

Показники стану ринку	Характеристика
Загальна потреба в продукції	Необхідна, але багатьма не признається (через фінансові вигоди)
Можливі річні обсяги випуску в натуральних показниках	До 500 копій
Ціна одиниці продукції	10$ (в комерційних цілях)
Річні обсяги випуску в вартісних показниках	100– 5000$
Динаміка ринку (якісна оцінка)	Зростає
Наявність обмежень для входу	Бажання розробників працювати лише над власним ПЗ, задля підтримки монополії у сфері
Показники стану ринку	Характеристика

продовження таблиці 5.4

Специфічні вимоги до стандартизації та сертифікації	Для ПЗ відсутні. Для коректної роботи - використання стандартів ISO 9126 та ISO 25010
Середня норма рентабельності в галузі (або по ринку)	78%

За попереднім оцінювання ринок не здається достатньо привабливим для входження. Але при проведенні збору статистичних даних, що свідчать про підвищення попиту на подібні розробки, через збільшення інтересу до постреляційних баз даних як з боку звичайних користувачів, так і з боку професійних робітників. Надалі визначаються потенційні групи клієнтів, їх характеристики, та формується орієнтовний перелік вимог до товару для кожної групи (таблиця 6.5).

Після визначення потенційних груп клієнтів проводиться аналіз ринкового середовища: складаються таблиці факторів, що сприяють ринковому впровадженню проекту, та факторів, що йому перешкоджають.
Результати представлені у таблицях 6.6 та 6.7 відповідно.
Після аналізу конкуренції проводиться більш детальний аналіз умов конкуренції в галузі (таблиця 6.9) - за моделлю п’яти сил М. Портера, яка вирізняє п’ять основних факторів, що впливають на привабливість вибору ринку з огляду на характер конкуренції:

- конкурент, що вже є у галузі;
- потенційні конкуренти;
- наявність товарів-замінників;
- постачальники, що конкурують за ринкову владу;
- споживачі, які конкурують за ринкову владу.

Таблиця 5.5 – Характеристика потенційних клієнтів стартап-проекту

Потреба, що формує ринок	Цільова аудиторія	Особливості поведінки споживачів	Вимоги споживачів до товару
Оцінка якості розрахунку показників	Філологи, що вивчають тексти та визначають авторів невідомих текстів	Розробники займаються написанням програм, які не завжди: відповідають стандартам , не завжди достатньо оптимізовані, що впливає на подальше життя створених програмних продуктів – виникають проблеми, недоліки та конфлікти. Тривале вирішення проблем несумісности або критичних помилок ПЗ.	– доступна ціна; – зручність і простота використання; – мобільність

Таблиця 5.6 – Фактори загроз

Фактор	Зміст загрози	Можлива реакція компанії
Поява конкурентів	Можлива поява конкурентів, які спроможуться створити більш якісний продукт. Можлива поява більш дешевих продуктів	Зменшення ціни з підвищенням якости при цьому, розробка удосконалень, розширення асортименту (додавання нових можливостей, нового функціоналу та/або додання можливостей розрахунку нових параметрів або можливість прогнозування стану кардіореспіраторної системи)
Зміни тенденцій ринку	Можлива ситуація, в якій з'явиться більш досконала програмна система від конкурентів, які значно довше на ринку.	Майже неможлива ситуація на найближчі багато років. Але можливості вирішення найпростіші - розробка нових сучасних необхідних удосконалень, тобто додання або заміна старого функціоналу на можливості розрахунку нових параметрів
Зниження репутації компанії	Можлива ситуація, коли конкуренти спроможуться на більший попит	Зміна партнерів, заключення нових контрактів, проведення рекламних та промо-акцій
Економічний спад	Відсутність попиту на товар компанії через економічну складову	Збільшення обсягів продажів, зменшення ціни; зміна цільової аудиторії

Таблиця 5.7 – Фактори можливостей

Фактор	Зміст загрози	Можлива реакція компанії
Невелика кількість конкурентів	На ринку на сьогоднішній день значна кількість конкурентів, проте їх програмні продукти в переважній більшості вузько спеціалізовані	Розповсюджувати створений продукт, розвивати його можливості
Відповідні тенденцій ринку	ІТ-ринок на сьогоднішній день потребує, а відповідно і надає всі можливості для впровадження систем, які надаватимуть користувачам можливість контролю систем організму, зокрема кардіореспіраторної.	Розповсюджувати створений продукт, розвивати його можливості
Можливість побудови власної репутації	Новий «гравець» на ринку має всі можливості для побудови власної репутації з «чистого листка»	Пошук замовників, можливих покупців створеного продукту, розширення бази замовників. Зарекомендувати себе, як надійну компанію. Можливо на вигідних умовах співпраці

Надалі проводиться аналіз пропозиції – визначаються загальні риси конкуренції на ринку (таблиця 5.8): визначаються тип можливої майбутньої конкуренції та її інтенсивність, рівень конкурентоспроможності за рівнем конкурентної боротьби, видами товарів і галузевою ознакою.

Таблиця 5.8 – Ступеневий аналіз конкуренції на ринку

Особливості конкурентного середовища	У чому проявляється дана характеристика	Вплив на діяльність підприємства (можливі дії компанії)
Тип конкуренції	Залежить від кількості конкурентів та якості надання ними послуг у порівнянні з послугами компанії	Покращення власного продукту через зниження ціни та підвищення якості
За рівнем конкурентної боротьби	Локальна Конкуренція на вітчизняному ринку	На вітчизняному ринку конкурентів мало, а тому встановлення власної бажаної ціни, та наробляти клієнтську базу. Перспектива – вихід на міжнародний рівень
За галузевою ознакою	Внутрішньогалузева Продукт націлений лише на конкретну сферу діяльності	Немає можливостей та сенсу розширювати функціонал за межі ІТ-сфери, але існує багато варіантів розвиватись всередині неї
Конкуренція за видами товарів	Марки-конкуренти Створений товар може мати конкурентів, які пропонують аналогічний товар	Зниження ціни, розширення функціональних, безплатне встановлення в державних закладах охорони здоров’я (задля популяризації методу)
Особливості конкурентного середовища	У чому проявляється дана характеристика	Вплив на діяльність підприємства (можливі дії компанії)

продовження таблиці 5.8

За характером конкурентних переваг	Цінова Важливо за скільки продається товар, та скільки з нього прибутку	Можливе підвищення ціни на нові розробки, зниження на старі версії для заохочення покупців у порівнянні з цінами конкурентів
За інтенсивністю	Марочна Можуть з'являтись конкуренти	На ринку цільової аудиторії поки що конкурентів не виявлено. Але при виході на міжнародний ринок потрібно рекламувати кращий функціонал створеного продукту, встановлювати конкурентоспроможні ціни, та доводити свою надійність

На основі аналізу конкуренції, проведеного у таблиці 5.9, а також із урахуванням характеристик ідеї проекту (таблиця 5.2), вимог споживачів до товару (таблиця 5.5) та факторів маркетингового середовища (таблиці 5.6 і 5.7) визначається та обґрунтовується перелік факторів конкурентоспроможності. Аналіз оформлюється за таблицею 5.9, обґрунтування факторів за таблицею 5.10.

За визначеними факторами конкурентоспроможності проводиться аналіз сильних та слабких сторін стартап-проекту, проведений у таблиці 5.11.

Таблиця 5.9 – Аналіз конкуренції в галузі за М.Портером

	Прямі конкуренти в галузі	Потенційні конкуренти	Клієнти	Товари-замінники
Складові галузі	Розробники аналогічних систем	Кращі продукти, менші ціни	Мають найбільше значення. Більш важлива їх кількість, аніж постійна співпраця	Відсутні. Є лише конкуренти аналогічних розробок
Висновки	Інтенсивність конкурентної боротьби з боку прямих конкурентів незначна	Наявні усі можливості входу на ринок. Потенційні конкуренти не виявлені. Строки виходу на ринок – один день	Необхідність клієнтської-бази, тому важливо знаходити можливості приваблення споживачів до власного продукту	Немає обмежень

Таблиця 5.10 – Обґрунтування факторів конкурентоспроможності

Фактор конкурентоспроможності	Обґрунтування
Невелика кількість конкурентів на ринку	На вітчизняному ринку, на який для старту націлена розроблена система, конкурентів мало
Доступність створеного продукту (програмно)	Немає жорстких системних вимог, програма буде працювати навіть на застарілих ПК
Фактор	Обґрунтування
Легкість і простота використання	Зручний зрозумілий інтерфейс, створені довідка та інструкція для користувача
Відсутня потреба у постійному супроводі	Не потребує супроводу спеціалістів і постійних доробок з боку розробника
Підключення до мережі Інтернет	Немає потреби у підключенні до мережі Інтернет після придбання продукту, на відміну від більшості аналогів

Таблиця **5**.11 – Порівняльний аналіз сильних та слабких сторін проекту

Фактор конкурентоспроможності	Бали 1-20	Рейтинг товарів-конкурентів						
		–3	–2	–1	0	+1	+2	+3
Мала кількість конкурентів	10				✓			
Системні вимоги	18				✓			
Простота використання	18	✓						
Не потрібен супровід	10					✓		

Фінальним етапом ринкового аналізу можливостей впровадження проекту є складання SWOT-аналізу (Strength, Weak, Opportunities, Troubles) (таблиця 5.12) на основі виділених ринкових загроз та можливостей, та сильних і слабких сторін.

Таблиця 5.12 – SWOT-аналіз проекту

Сильні сторони (S):	Слабкі сторони (W):
– висока точність індентифікації; – розвинута кастомізація; – гнучка політика керівництва; – інноваційні технології	– брак власного устаткування; – складність розробки; – недостатньо оборотних коштів; – відсутність репутації компанії;
Можливості (O):	Загрози (T):
– розширення сфер застосування; – додаткові методи класифікації; – вихід на нові ринки; – розширення клієнтської бази; – співробітництво з іншими компаніями	– поява нових конкурентів; – зміни тенденцій попиту; – зниження репутації компанії; – економічний спад

На основі SWOT-аналізу розробляються альтернативи ринкової поведінки (перелік заходів) для виведення стартап-проекту на ринок та орієнтовний оптимальний час їх ринкової реалізації з огляду на потенційні проекти конкурентів, що можуть бути виведені на ринок.

Визначені альтернативи аналізуються з точки зору строків та ймовірності отримання ресурсів (таблиця 6.13).

Таблиця 5.13 – Альтернативи ринкового впровадження стартап-проекту

Альтернатива ринкової поведінки	Ймовірність отримання ресурсів	Строки реалізації
Вихід на нові ринки	Пошук інвесторів	1-6 місяців
Розширення виробничої лінії	Пошук інвесторів	Після виходу на ринок основного продукту, до 6 місяців

Спочатку потрібно вивести на основний ринок розроблену систему, а вже потім шукати можливості розширення програмного функціоналу для користувачів.

5.4 Розроблення ринкової стратегії

Розроблення ринкової стратегії першим кроком передбачає визначення стратегії охоплення ринку: опис цільових груп потенційних споживачів, які визначені у таблиці 5.14.

Таблиця 5.14 – Вибір цільових груп потенційних споживачів

Опис цільової групи потенційних клієнтів	Готовність споживачів сприйняти продукт	Орієнтовний попит в сегменті	Інтенсивність конкуренції в сегменті	Простота входу у сегмент
Філологи	Потребують	Попит є	Незначна	Просто
Студенти	Потребують	Попит є, проте нижчий ніж у філологів	Незначна	Помірно

Які цільові групи обрано: оскільки різниця між цільовими групами зовсім незначна, а також враховуючи той факт, що компанія має бажання почати продажі (а відповідно і отримання прибутку) якомога швидше, то доцільно враховувати обидві цільові групи, тобто використовувати масовий маркетинг, пропонуючи стандартизовану програму.

За результатами аналізу потенційних груп споживачів (сегментів) автори ідеї обирають цільові групи, для яких вони пропонуватимуть свій товар, та визначають стратегію охоплення ринку.

Для роботи в обраних сегментах ринку необхідно сформувати базову стратегію розвитку, яка визначається у таблиці 5.15.

Вибір стратегії конкурентної поведінки визначається у таблиці 5.16.

Таблиця 5.15 – Визначення базової стратегії розвитку

Обрана альтернатива розвитку проекту	Стратегія охоплення ринку	Ключові конкурентоспроможні позиції відповідно до обраної альтернативи	Базова стратегія розвитку
Вихід на нові ринки	Стратегія диференціації	Надання товару відмінних якостей, які роблять систему особливою на фоні аналогічних розробок	Стратегія диференціації
Розширення виробничої лінії	Стратегія диференціації (допускається стратегія спеціалізації)	Надання товару кращих властивостей	Стратегія диференціації (допускається стратегія спеціалізації)

Таблиця 5.16 – Визначення базової конкурентної поведінки

Чи є проект «першопроходцем» на ринку	Так
Чи буде компанія шукати нових споживачів, або забирати існуючих у конкурентів?	Обидва варіанти
Чи буде компанія копіювати основні характеристики товару конкурента, і які?	Ні
Стратегія конкурентної поведінки	Стратегія виклику лідера

На основі вимог споживачів з обраних сегментів до постачальника (стартап-компанії) та до продукту, а також в залежності від обраної базової стратегії розвитку та стратегії конкурентної поведінки розробляється стратегія позиціонування (таблиця 6.17), що полягає у формуванні ринкової позиції

(комплексу асоціацій), за яким споживачі мають ідентифікувати торгівельну марку або проект.

Таблиця 5.17 –Визначення стратегії позиціонування

Вимоги до товару цільової аудиторії	Базова стратегія розвитку	Ключові конкурентоспроможні позиції власного стартап-проекту	Вибір асоціацій, які мають сформувати комплексну позицію власного проекту
Доступна ціна, простота і зручність використання, універсальність	Стратегія диференціації	Вирішення важливих поставлених задач швидко, легко та зрозуміло навіть без інструкцій. Легкість і простота у використанні. Доступність через ціну та технічні характеристики	– стандарти якості – метрики ПЗ – ASQAS - automated system of quality assessment software – Холстед, LOC, Джилб, МакКейб

Результатом є узгоджена система рішень щодо ринкової поведінки стартап-компанії, яка визначатиме напрями роботи стартап-компанії на ринку.

Отже, робота стартап-компанії на ринку повинна бути спланована орієнтовано таким чином: за стратегією диференціації виконаний і буде поширюватись товар відмінний за властивостями від своїх аналогів, дотримуючись у конкурентній поведінці стратегії «виклику лідера», тобто випускається один товар для усіх можливих споживачів.

Надалі розроблена трирівнева маркетингова модель товару: уточняються ідея продукту, його фізичні складові, особливості процесу його надання (таблиця 5.19).

5.5 Розроблення маркетингової програми стартап-проекту

Першим кроком є формування маркетингової концепції товару, який отримає споживач. Для цього у таблиці 6.18 підсумовані результати попереднього аналізу конкурентоспроможності товару.

Таблиця 5.18 – Визначення ключових переваг концепції потенційного товару

Потреба	Вигода, яку пропонує товар	Ключові переваги перед конкурентами
Оцінка якості ПП	Оцінка за 4 метриками. Удосконалення оцінки будь-якої з обраних характеристик.	Розрахункові показники, точність та достовірність яких можна оцінювати; кількість вхідних параметрів; самостійність програмної системи.

Таблиця 5.19 – Опис трьох рівнів моделі товару

Рівні товару	Сутність та складові
Товар за задумом	Механізм атрибуції тексту на основі стильометричних особливостей авторів та за допомогою нейронної мережі буде визначати автора ще невідомого програмі тексту
Товар у реальному виконанні	Властивості/характеристики
	Реалізовано систему визначення автора невідомого тексту. Реалізована система пошуку стилістичних особливостей автора. Реалізовано графічне представлення результатів. Доведена адекватність розрахунків.
	Сутність та складові

продовження таблиці 5.19

	Якість: тестування пройшло задовільно
	До продажу: стандартна розроблена система (модуль «Стильометрія» та модель «Визначення автора невідомого тексту»)
	Після продажу: додані додаткові можливості, збільшення споживчої бази

За рахунок чого потенційний товар буде захищено від копіювання: від копіювання потенційний товар захистити не складає проблеми. Розроблена математични модель, на якій базується програмна система, публікувалась лише у загальних рисах, а без математичної моделі цей ПП лише набір рядків коду. Але створення комплексної математичної моделі кардіореспіраторної системи людини у створеному проекті є науковою новизною; не реалізовувалось раніше, а тому є необхідність у фіксуванні авторських прав або отриманні патенту.

Визначення цінових меж, якими необхідно керуватись при встановленні ціни на потенційний товар, яке передбачає аналіз ціни на товари-аналоги, а також аналіз рівня доходів цільової групи споживачів описано в таблиці 6.20.

Таблиця 5.20 – Визначення меж встановлення ціни

Рівень цін на товари-аналоги	Рівень доходів цільової групи споживачів	Верхня та нижня межі встановлення ціни
15 – 500 $	500 – 5000 $	20 – 50 $

Наступним кроком є визначення оптимальної системи збуту, в межах якого приймається рішення (таблиця 5.21): проводити збут власними силами або залучати сторонніх посередників, вибір та обґрунтування оптимальної глибини каналу збуту, вибір та обґрунтування виду посередників.

Таблиця 5.21 – Формування системи збуту

Специфіка закупівельної поведінки цільових клієнтів	Функції збуту, які має виконувати постачальник товару	Глибина каналу збуту	Оптимальна система збуту
Бажання отримати більше за менші гроші	Пошук клієнтської бази та продаж	Нульовий рівень: тільки виробник	Вертикальна маркетингова система

Останньою складової маркетингової програми є розроблення концепції маркетингових комунікацій, що спирається на попередньо обрану основу для позиціонування, визначену специфіку поведінки клієнтів (таблиця 5.22).

Таблиця 5.22 – Концепція маркетингових комунікацій

Поведінка цільових клієнтів	Канали комунікацій цільових клієнтів	Ключові позиції, обрані для позиціонування	Завдання рекламного повідомлення
Бажання отримати більше за менші гроші	Будь-які	Низька ціна Широкий вибір функціоналу Легкий і простий у використанні продукт	Донести до користувача суть продукту, його якість, та залучити якомога більше зацікавлених клієнтів

5.6 Висновки

В цьому розділі було проведено маркетинговий аналіз з метою визначення можливості та доцільності ринкової комерціалізації проекту програмного забезпечення визначення автора художнього тексту.

Результати дослідження свідчать про можливість ринкової комерціалізації, що обґрунтовується позитивною динамікою нового, ще не до кінця сформованого, ринку, потенціал якого досить значний, судячи з західних більш розвинутих регіонів світу.

При комерційній реалізації проекту можуть стати на заваді економічне та правове становище в країні.

При побудові маркетингової компанії варто спиратися на прямий канал збуту нульового рівня, та висвітлювати ефективність рішень такого роду, унікальність рішення та значущість впровадження для встановлення конкурентної переваги, а також на швидке та легке впровадження та інтеграцію.

Підсумок маркетингового аналізу вказує на доцільність подальшої реалізації проекту.

ВИСНОВКИ

В даній магістерській роботі було розглянуто опис загальних теоретичних аспектів визнаяенна автора тексту. Були описані визначення, характеристики та властивості визначення авторського стилю, а також розглянуті основні підходи до їх розв’язання.

Було описанао та розроблено методику визначення автора тексту на основі синтаксичних та лексичних текстових ознак. Описано методологію навчання текстового класифікатора, пошуку оптимальної кількості структурних одиниць, зокрема кількості прихованих шарів та виходів нейроної мережі. Навчання та параметрів регуляризації. Розглянуто способи оцінки побудованої моделі та представлена загальна архітектура нейроної мережі.

Наступним було складено та описано функціональні та нефункціональні вимоги до програмного забезпечення. Також описано базову схему роботи програмного забезпечення.

В розділі 4 було досліджено вплив параметрів моделі на результати класифікації. Проведені результати залежності збільшення кількості навчальних даних на ефективність класифікатора. Також було показано, що деякі роботи неправильно класифіковані, оскільки текстові функції, що описують їх, недостатньо точні для виконання завдання.

Також було проведено маркетинговий аналіз з метою визначення можливості та доцільності ринкової комерціалізації проекту програмного забезпечення визначення автора художнього тексту.

СПИСОК ЛІТЕРАТУРИ

1) Herdan Gustav. The advanced theory of language as choise and chance. Kommunikation und Kybernetik in Einzeldarstellungen. – 1966. – С. 14–437.

2) Общий взгляд на машинное обучение: классификация текста с помощью нейронных сетей и TensorFlow URL: Режим доступу: https://tproger.ru/translations/text-classification-tensorflow-neural-networks. – (дата звернення 03.07.2019).

3) Mendenhall T. C. A mechanical solution of a literary problem. The Popular Science Monthly, c.97-105. URL: https://en.wikisource.org/wiki/Popular_Science_Monthly/V olume_60/Decem ber_1901/A_Mechanical_Solution_of_a_Literary_Problem.

4) Juola P. Authorship attribution. Foundations and Trends in Information Retrieval. – 2006. – No 1. – С. 233—334. – URL: https://books.google.com.ua/books/about/Authorship_Attribution.html?id=_B 2zDLdqe60C&redir_esc=y.

5) Bobadilla J., Ortega F., Recommender systems survey .Knowledge-Based Systems – 2013. – pp. 109-132.

6) Stamatatos E. Author identification: Using text sampling to handle the class imbalance problem. Information Processing and Management: an International Journal. – 2008. – No 44. – С.790–799. URL: https://dl.acm.org/citation.cfm?id=1347518.

7) Quantitative authorship attribution: An evaluation of techniques [Електронний ресурс] / Grieve J. // Literary and Linguistic Computing. – 2005. – No22. – С.251-270. URL: https://academic.oup.com/dsh/article-abstract/22/3/251/951481?redirectedFrom=fulltext. – (дата звернення 01.04.2018). – Назва з екрана.

8) Experiments with mood classification in blog posts [Електронний ресурс] / Mishne G. // 1st workshop on stylistic analysis of text for information access. – 2005.URL:

http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.111.2693&rep=rep1&type=pdf. – (дата звернення 10.04.2018). – Назва з екрана.

9) Agrawal R., Rajagopalan S., Srikant R., Xu Y. Mining newsgroups using networks arising from social behavior. Proceedings of the 12th international conference on World Wide Web. – 2003. – С. 529–535. – URL: https://dl.acm.org/citation.cfm?id=775227. –

10) Naive Bayes classifier/. URL: https://en.wikipedia.org/wiki/Naive_Bayes_classifier. – (дата звернення 02.03.2019).

11) Support vector machine URL: https://en.wikipedia.org/wiki/Support_vector_machine. – (дата звернення 02.03.2019).

12) Николенко С. Рекомендательные системы: теорема Байеса и наивный байесовский классификатор. Блог компании Surfingbird. URL: https://habr.com/company/surfingbird/blog/150207/ – (дата звернення 18.07.2019).

13) Николенко С. Рекомендательные системы: SVD, часть I. URL: https://habr.com/company/surfingbird/blog/139863/ – (дата звернення 17.08.2019).

14) Олег Храпов. Визначення автора тексту з використанням ANN//Наука онлайн: Міжнародний електронний науковий журнал - 2019. - №12. URL: https://nauka-online.com/ua/publications/ informatsionnye-tehnologii/2019/12/viznachennya-avtora-tekstu-z-vikoristannyam-ann/

15) Воронцов К. В. Математические методы обучения по прецедентам (теория обучения машин). URL: http://www.machinelearning.ru/wiki/images/6/6d/Voron-ML-1.pdf. – (дата звернення 11.06.2019).

16) Zheng, R., Chen, H., and Huang, Z. (2006). A framework for author-ship identification of online messages: Writing-style features and classification techniques. №57(3):378–393.

17) Олег Храпов. Визначення статі автора короткого тексту методами машинного навчання. Міжнародний електронний науковий журнал - 2019. - №11. URL: https://nauka-online.com/ua/publications/tehnicheskie-nauki/2019/11/opredelenie-pola-avtora-korotkogo-teksta-metodami-mashinnogo-obucheniya/

ДОДАТОК А РЕЗУЛЬТАТИ КЛАСИФІКАЦІЇ

Таблиця А.1 – Точність класифікації для 5 робіт на автора

Кількість авторів	Accuracy	F - міра	Капа статистика
2	100	100	100
3	93	93	90
4	85	85	80
5	84	83	80
6	63	61	56
7	66	65	61
8	63	62	58
9	59	60	55
10	48	48	42
11	47	46	42
12	53	54	49
13	55	56	51
14	45	45	41
15	46	46	42
16	40	41	37
17	46	47	43
18	45	45	42
19	43	44	41
20	42	43	40

Таблиця А.2 – Точність класифікації для 10 робіт на автора

Кількість авторів	Accuracy	F - міра	Капа статистика
2	100	100	100
3	93	93	90
4	86	87	82
5	89	90	87
6	81	81	77
7	86	87	85
8	78	78	75
9	77	77	74
10	82	82	81
11	77	78	76
12	77	77	75
13	68	70	66
14	71	72	69
15	68	69	66
16	68	68	66
17	65	66	64
18	70	71	69
19	67	68	66
20	67	68	66

Таблиця А.3 – Точність класифікації для 15 робіт на автора

Кількість авторів	Accuracy	F - міра	Капа статистика
2	96	97	93
3	84	85	77
4	81	82	75
5	74	74	68
6	68	68	62
7	75	75	72
8	74	74	71
9	63	63	59
10	56	56	52
11	62	62	58
12	61	61	58
13	56	57	53
14	58	59	55
15	56	56	54
16	53	53	50
17	58	58	56
18	53	52	51
19	52	52	50
20	52	52	50

Таблиця А.4 – Точність класифікації для 20 робіт на автора

Кількість авторів	Accuracy	F - міра	Капа статистика
2	95	95	90
3	95	95	93
4	87	87	83
5	82	83	79
6	80	80	77
7	81	81	78
8	81	82	79
9	77	78	75
10	73	74	70
11	68	68	65
12	67	67	64
13	67	68	65
14	66	67	64
15	64	65	62
16	63	63	61
17	62	63	61
18	68	68	66
19	59	59	57
20	61	61	60

ДОДАТОК Б ВИКОРИСТАНІ ТЕКСТИ

Таблиця Б.1 – Використані тексти

№	Автор	Тексти для тренування	Тексти для тестування
1	Іван Франко	«Борис Граб», «Довбанюк», «Борислав», «Абу-Касимові капці», «Веснянки», «В тюремнім шпиталі», «Вільгельм Телль», «Вовк-старшина», «Вугляр», «Гава», «Гава і Вовкун», «Гадки на межі», «Геній», «Герой поневолї», «Грицева шкільна наука», «Давнє й нове», «Два приятелі», «Декадент», «З бурливих літ», «Захар Беркут»	«Звірячий буджет», «Івась Новітний», «Іригація»
2	Остап Вишня	«А народ воювати не хоче», «Бекас», «Бенгальський тигр», «Біля річки», «Вальдшнеп», «Василь Іванович», «Васько, небіж мій», «Ведмідь», «Велікі ростіть», «Дика коза», «Геометрія», «Дилда», «Дідів прогноз», «Драстуйте», «Дрохва», «Екіпіровка мисливця», «З крякухою на озері», «Зенітка», «Капітан і гарпунник», «Лебідь».	«Лицем до села», «Літературні шаржі», «Лось», «Мистецькі силуети».

продовження таблиці Б.1

3	Володимир Виниченко	«Memento», «Антерпреньор Гаркун-Задунайський», «Бабусин подарунок», «Базар», «Біля машини», «Божки», «Брехня», «Голод», «Голос», «Дим», «Дочка жандарма», «Заручини», «Зіна», «Контрасти», «Краса і сила», «Купля», «Між двох сил», «Молода кров», «Момент», «На пристані».	«Намисто», «Роботи!», «Салдатики!», «Промінь сонця».
4	Марко Вовчок	«Викуп», «Гайдамаки», «Горпина», «Данило Гурч», «Два сини», «Дев'ять братів і десята сестриця Галя», «Дяк», «Затейник», «Игрушечка», «Інститутка», «Кармелюк», «Катерина», «Козачка», «Купеческая дочка», «Ледащиця», «Лемеривна», «Максим Гримач», «Маруся», «Маша», «Народні оповідання»,	«Не до пари», «Невільничка», «Одарка», «Отець Андрій», «Павло Чорнокрил».

продовження таблиці Б.1

5	Павло Загребельний	«В-ван!», «Вигнання з раю», «Волосожар», «Гопак під шибеницею», «День для прийдешнього», «День шостий», «Диво», «Добрий диявол», «Дума про невмирущого», «Дух Чингісхана», «Євпраксія», «З погляду вічності», «Ключ від сейфа», «Коров'ячий детектив», «Левине серце», «Марево», «Меланія», «Андрофонос», «Неймовірні оповідання», «Неложними устами»	«Покорчене озеро», «Попіл снів», «Рефлективне управління», «Розгін», «Роксолана»
6	Григір Тютюник	«Азарт», «Біла мара», «Бовкун», «Бушля», «В сутінки», «Василь Васильович», «Віктор», «Вогник далеко в степу», «Вуточка», «Гвинт», «Грамотний», «Громовик», «Груші з копанки», «День мій суботній», «Деревій», «Дивак», «Дикий», «Додому, додому...», «Дядько Никін», «Забутий курінь».	«Іван Срібний» «Кленовий пагін», «Климко», «Колиска», «Комета».

продовження таблиці Б.1

7	Микола Хвильовий	«Арабески», «Бандити», «Бараки, що за містом», «В електричний вік», «Вальдшнепи», «Вступна новела», «Дорога й ластівка», «Думки проти течії», «Елегія», «Життя», «З лабораторії», «Зав'язка», «Заулок», «Іван Іванович», «Із Вариної біографії», «Кімната ч. 2», «Кіт у чоботях», «Клавіатурте», «Колонії, вілли...», «Легенда».	«На глухім шляху», «На озера», «Наречений», «Свиня».
8	Іван Нечуй-Левицький	«Апокаліпсична картина в Києві», «Афонський пройдисвіт», «Баба Параска та баба Палажка», «Без пуття», «Біда бабі Палажці Солов'їсі», «Біда бабі Парасці Гришісі», «Бідний думкою багатіє», «Бурлачка», «В диму та в полум'ї», «В Карпатах», «Вечір на Владимирській горі», «Вітрогон», «Вольне кохання», «Гастролі», «Гетьман Іван Виговський», «Голодному й опеньки – м'ясо», «Два брати», «Дві милі», «Дві московки», «Дивовижний похорон».	«Дрегочин та Остріг», «Дрібна рибка», «Єврейський Скнеря», «Живцем поховані».

продовження таблиці Б.1

9	Анатолій Дімаров	«Артистка», «Баба Ониська і четверо її чоловіків», «Бабуля», «Бен Ладен і Буш», «Біль і гнів», «Біля люстра», «Блакитна дитина», «Боги на продаж», «Борьчин хазяїн», «Вампір», «Вершини», «Випадок у лікарні», «Випив і закусив», «Вони такі!», «Входження в ринок», «Галя дивиться "Багатих..."», «Голгофа», «Гонорар», «Гусятниця», «Дід Черепок косить пшеницю».	«Діти», «Для чого людині серце», «До копійки копієчка», «Друга планета», «Другий кухоль», «З вітерцем», «За Шекспіром», «Зарізали півня», «Зурочили»
10	Леся Українка	«Contra spem spero!», «Fait Nox!», «To be or not to be?..», «Ангел помсти», «Біда навчить», «Бояриня», «В катакомбах», «Веснянка», «Вечірня година», «Віла-посестра», «Все, все покинуть, до тебе полинуть...», «Грішниця», «Давня весна», «Давня казка», «Дим», «До мого фортепіано», «Досвітні огні», «Колискова», «Конвалія», «Королівна».	«На руїнах», «Напис в руїні», «Одержима», «Одно слово», «Оргія».

продовження таблиці Б.1

11	Михайло Коцюбинський	«Fata Morgana», «Intermezzo», «Persona grata», «Брати-місяці», «В дорозі», «В путах шайтана», «Відьма», «Він іде!», «Дебют», «Десять робітників», «Для загального добра», «Дорогою ціною», «З глибини», «Івасик та Тарасик», «Коні не винні», «Лист», «Листи до Олександри Аплаксіної», «Лялечка», «Маленький грішник», «На віру».	«На камені», «На крилах пісні», «На острові», «Невідомий», «Нюренберзьке яйце».
12	Олександр Довженко	«Аероград», «Антарктида», «Арсенал», «Бронза», «Відступник», «Воля до життя», «Життя в цвіту», «Зачарована Десна», «Земля», «Капітан Ус», «Китайський святий», «Корінь життя», «Мати», «Мічурін», «На колючому дроті», «Невідомий», «Незабутнє», «Ніч перед боєм», «Перемога», «Повість полум'яних літ».	«Поема про море», «Потомки запорожців», «Прощай, Америко!», «Сіятель»

продовження таблиці Б.1

13	Ліна Костенко	«Білочка восени», «Біль єдиної зброї», «Божевілля моє, божемилля…», «Бузиновий цар», «В дні, прожиті печально і просто…», «В маєтку гетьмана Івана Сулими...», «В пустелі сизих вечорів…», «Ван-Ґоґ», «Веселий дощ», «Вечірнє сонце, дякую за день!», «Любов Нансена», «Любов Потьомкіна», «Люди з Табулена», «Мадам Андро», «Майже переклад з провансальської», «Мало всього — ще і тугу цю вовчу…», «Маруся Чурай», «Мати», «Мені відкрилась істина печальна...».	«Між іншим», «Скіфська баба», «Смертельний падеграс», «Сміх», «Сніг у Флоренції», «Сольфеджіо».

продовження таблиці Б.1

14	Юрій Андрухович	«Пісня мандрівного спудея», «Про дівчат. Вони тут не стільки юні...», «Рекреації», «Різдвяні вакації», «Серпень.Дністер», «Скупа природа наших середмість...», «Спокушання», «Футбол на монастирському подвір'ї», «Час і місце, або Моя остання територія», «Shevchenko is OK», «Аве, "Крайслер"!», «Астролог», «Весна виникала, де тільки могла...», «Вольф Мессінґ. Вигнання голубів», «Вступ до географії», «Дванадцять обручів», «Зима і сни вартового», «Козак Ямайка», «Московіада», «Я заліз у тугу, як в тогу чи в робу».	«Як ми вбили Пятраса», «Ніжність», «Спокушання».
15	Іван Драч	«Балада про Сар'янів та Ван-Гогів», «Балада про соняшник», «Балада про ступу», «Балада про усмішку», «Балада роду», «Балада творчості», «Баллада о моем осколке», «Баляда скромності», «Барвінок», «Бас і плач», «Врубелівський етюд», «Гітара Пабло	«Матері», «Мачинка», «Монтень чи Свіфт», «Ніж у Сонці».

		Неруди», «Грузинській дівчині, убитій саперною лопаткою», «Дві сестри», «Десь на дні моїх ночей...», «Дід Любимененепокинь», «Етюд кохання», «Етюд про хліб», «Жартівлива балада про випрані штани», «Жінки і лелеки».	
16	Валерій Шевчук	«Бігунець та Котило», «Білецькі», «Біс плоті», «Вони завжди були разом», «Вулиця», «Горбунка Зоя», «Двері навстіж», «Двоє на березі», «Дерево пам'яті», «Дзеркало», «Освітлена сонцем кімната», «Останній день», «Павло-диякон», «Панна квітів», «Під синичий подзвін», «Поглинач запахів», «Постріл», «Початок жаху», «Привид мертвого дому», «Птахи з невидимого острова».	«Роман юрби», «Сивий», «Сиві хмари», «Смуга нещасть», «Сон сподіваної віри».

продовження таблиці Б.1

17	Тарас Шевченко	«Автобіографічний нарис», «Блажений муж на лукаву...», «Близнецы», «Варнак», «Великий льох, «Відьма», «Гайдамаки», «Гамалія», «Гоголю», «Готово! Парус розпустили...», «Назар Стодоля», «Наймичка», «Не гріє сонце на чужині...», «Не нарікаю я на бога...», «Не тополю високую...», «Неофіти», «О думи мої! О славо злая!..», «Один у другого питаєм…», «Ой три шляхи широкії...», «Перебендя».	«Чигрине, Чигрине...», «Щоденник», «Я не нездужаю, нівроку...», «Як маю я журитися...», «Якби ви знали, паничі ...».
18	Валер’ян Підмогильний	«Історія пані Євги», «Комуніст», «Місто», «На іменинах», «На селі», «Невеличка драма», «П'ятдесят верстов», «Повість без назви», «Повстанці», «Проблема хліба», «В епідемічному бараці», «Важке питання», «Ваня», «Військовий літун», «Гайдамака», «Дід Яким», «Добрий Бог», «З життя будинку», «За день», «Іван Босий».	«Історія пані Євги», «Комуніст», «Місто», «На іменинах», «На селі».

продовження таблиці Б.1

19	Олесь Гончар	«Дніпровський вітер», «Дорога за хмари», «Дядько Роман і золотокрилки», «Жайворонок», «З тих ночей», «За мить щастя», «Завжди солдати», «Залізний острів», «Земля гуде», «Зірниці», «На косі», «Народний артист», «Нехай живе життя», «Ніч мужності», «Ода тій хаті, що в снігах», «Орхідеї з тропіків», «Партизанська іскра», «Перекоп», «Під далекими соснами», «Пізнє прозріння».	«Ілонка», «Корида», «Крапля крові», «Кресафт», «Літньої ночі».
20	Костецький Анатолій	«Бурульки», «Бюро знахідок», «Велосипед», «Веселий сніг», «Весна», «Весняні дарунки», «Відчинене вікно», «Вірш без кінця», «Вірш-хатка», «Вітя може все!», «Не хочу», «Невсидющий Василь», «Нема нікого вдома», «Неуважна сороконіжка», «Нове життя», «Осінній дощ», «Паперовий змій та я», «Півники», «Пісня для всіх», «Поговорили...».	«Про друга», «Про що виспівує струмок», «Проста арифметика», «Радісна квітка».

ДОДАТОК В СТУКТУРА ARFF ФАЙЛУ

Нижче наводиться приклад вигляду файлу в форматі ARFF.

```
@ATTRIBUTEtotalNumberOfWords    NUMERIC
@ATTRIBUTEtotalNumberOfPunctuationMarks    NUMERIC
@ATTRIBUTEaverageFunctionalWords    NUMERIC
@ATTRIBUTEaverageWordLength    NUMERIC
@ATTRIBUTEclass    {Костенко,Франко}

@DATA
3987,332,317.2,5.6,Костенко
4001,221,416.1,4.7,Костенко
4012,222,519.3,5.1,Костенко
2081,317,44.5,6.2,Франко
1873,220,254.4,5.2,Франко
2530,229,421.7,5.4,Франко
```

ДОДАТОК Г ЛІСТНИНГ ПРОГРАМИ

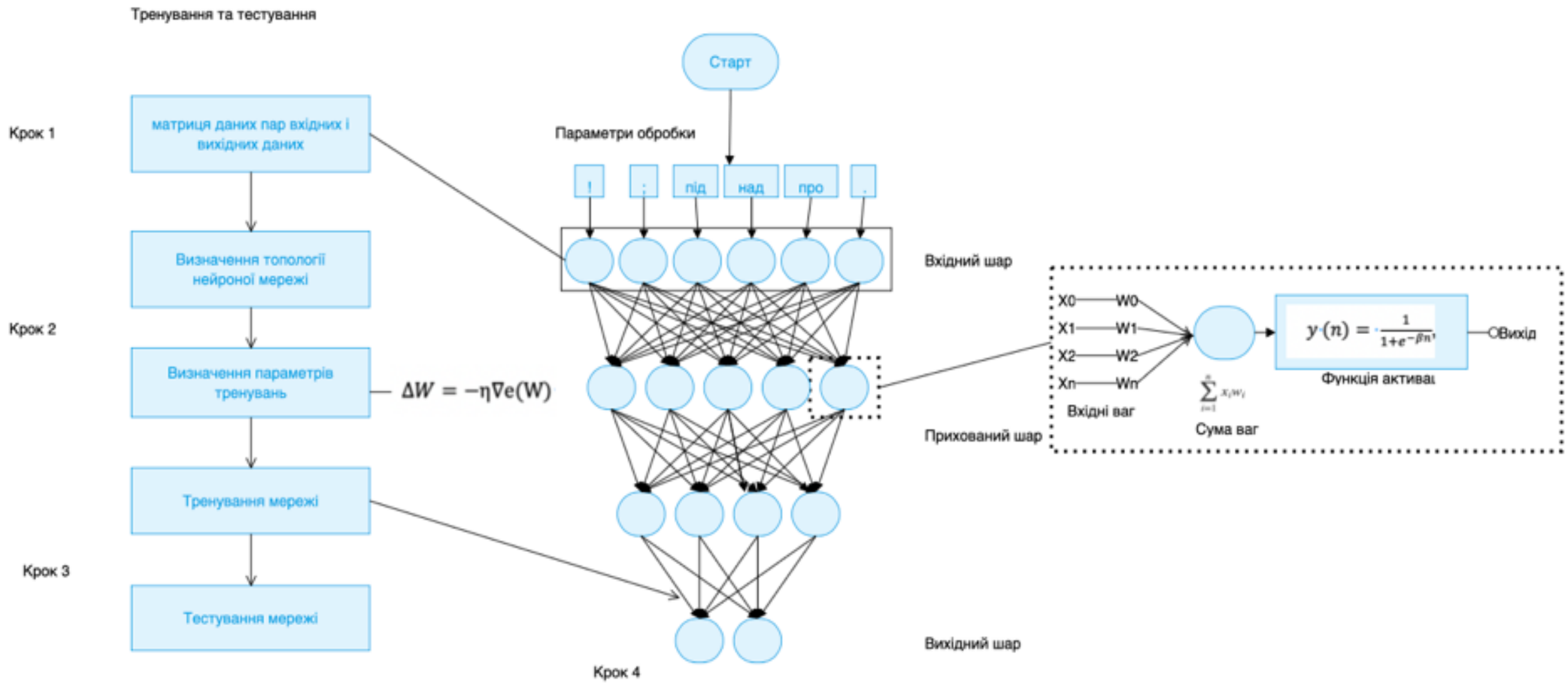
```
public static List<List<Integer>> confusionM = null;
    public static List<List<Integer>> getMatrix(){       if
(confusionM != null)          return confusionM;
else          throw );    }             public static
List<TextData> parseAll(List<Article> artArray) {
initMatrix();       ArrayList<TextData> textsData =
new ArrayList<TextData>();       HashMap<String,
AuthorData> authData = new HashMap<String,
AuthorData>();       for (Article article : artArray){
textsData.add(parse(article));       }       return
textsData;    }

public static TextData parse(Article article) {
          byte[] bytes = null;
          int paragraphs = 1;
          HashMap<Character, Integer> punctuation
= new HashMap<Character, Integer>()
          { {';', 0}, {'!', 0}, {'?', 0}, {':', 0}, {',', 0}, {'.', 0},
{'"', 0}, {'-', 0} };

          try {
                    bytes =
Files.readAllBytes(Paths.get(article.getCurrent().getP
ath()));
          } catch (IOException e1) {
                    e1.printStackTrace();
          }
          String text = new String(bytes,
StandardCharsets.UTF_8);
          for(int i = 0; i < text.length(); i++) {
                    Character c = text.charAt(i);
                    if(punctuation.containsKey(c)) {

          punctuation.put(text.charAt(i),
punctuation.get(c) + 1);
                    }
          }

          String[] lines = text.split("\n");
          ArrayList<Integer> numberOfWords = new
ArrayList<Integer>();
          for (String line : lines) {
                    numberOfWords.add(line.split("
").length);

          }
          return new TextData(article.getAuthor(),
punctuation, numberOfWords,  text.length(),
lines.length);
}
private static void initMatrix() {
          System.out.println("Initialisation de la
matrice de confusion");
          //if (confusionM == null){
                    int sizeee =
ListBuilder.authorList().size();
                    confusionM = new
ArrayList<List<Integer>>(sizeee);
                    for(int i=0; i<sizeee; ++i){
                              ArrayList<Integer> tmpl =
new ArrayList<Integer>();
                              for (int j = 0; j < sizeee;
++j){
                                        tmpl.add(j, 0);
                              }
                              confusionM.add(i, tmpl);
                    }
          //}
}

public static boolean isTitle(String line) {
          return line.equals(line.toUpperCase());
}

public static void testAll(List<TextData> trainingSet,
List<Article> testSet) {

          resetMatrix();
          int k =  10;
          int goodAnswers = 0;
          int yolorandom = 0;
          for (Article art : testSet) {
                    Entry<String, Integer> res =
KNearestNeighborsClassifier.getResponse(KNearest
NeighborsClassifier.getNeighbors(trainingSet,
parse(art), k));

                    String realAuthor = art.getAuthor();
                    Integer idRealAuthor =
ListBuilder.authorList().get(realAuthor);

                    String computedAuthor =
res.getKey();
                    Integer idComputedAuthor =
ListBuilder.authorList().get(computedAuthor);


          confusionM.get(idRealAuthor).set(idCompu
tedAuthor,
confusionM.get(idRealAuthor).get(idComputedAuth
or)+1);

                    if
(realAuthor.equals(computedAuthor)) {
                              goodAnswers++;
                              if (res.getValue() == 1){
                                        yolorandom++;
                              }
                    }
public static void main(String [] args) {
```

```
		List<Article> trainSet=
ListBuilder.buildList("../../Data/Reuters50_50/C50tra
in/");
		List<TextData> trainingSet =
parseAll(trainSet);
		TextData.setMinMaxFeatures(trainingSet);

public class TextData {
		public static TextData max;
		public static TextData min;

		public String author;
		public HashMap<Character, Integer>
punctuation;
		public HashMap<String, Double> features;
		public double lineLength;
		public int textSize;
		public int nbParagraphs;
		public int nbLines;
		public double nbWords;

		public TextData(String author) {
			this.author = author;
			features = new HashMap<String,
Double>();
		}

		public TextData(String author,
HashMap<Character, Integer> punct, List<Integer>
linesLengths, int paras, int size, int nbLines, int
mName, int abrev) {
			this.author = author;
			features = new HashMap<String,
Double>();
			features.put("lineLengths",
ListCalc.median(linesLengths));
			features.put("nbParagraphs",
(double) paras);
			features.put("textSize", (double)
size);
			features.put("nbWords", (double)
linesLengths.stream().mapToInt(Integer::intValue).su
m() / linesLengths.size());
			features.put("nbLines", (double)
nbLines);
			features.put("mName", (double)
mName);
			features.put("abrev", (double)
abrev);
			//punctuation = new
HashMap<Character, Integer>(punct);
			for (Character s : punct.keySet()) {

		features.put("punctuation"+ s, (double)
punct.get(s));
			}
			//lineLength =
ListCalc.median(linesLengths);
			//nbParagraphs = paras;
			//textSize = size;
			//nbWords =
linesLengths.stream().mapToInt(Integer::intValue).su
m() / linesLengths.size();
			//this.nbLines = nbLines;
		}

		public ArrayList<Double> getVals(){
			return new ArrayList<Double>() {{
				//add((double) textSize);
				add(lineLength);
add((double) nbParagraphs); add((double) nbLines);
				add((double)
punctuation.get(';')); add((double)
punctuation.get(',')); add((double)
punctuation.get('?'));
				add((double)
punctuation.get('!')); add((double)
punctuation.get(':')); add(nbWords);
			}};
		}
		public static void
setMinMaxFeatures(List<TextData> ld) {
			if (ld.size() == 0) {
				return;
			}
			else if (ld.size() == 1) {
				max = ld.get(0);
				min = ld.get(0);
				return;
			}
			max = new TextData("max");
			min = new TextData("min");
			Set<String> keys =
ld.get(0).features.keySet();
			for (TextData d : ld) {
				for (String feature : keys) {
					double curVal =
d.features.get(feature);
					if
(!max.features.containsKey(feature) ||
max.features.get(feature) < curVal) {

		max.features.put(feature, curVal);
					}
					if
(!min.features.containsKey(feature) ||
min.features.get(feature) > curVal) {

		min.features.put(feature, curVal);
					}
				}
			}
		}
		public double getNormalized(String
feature){
			return (this.features.get(feature) -
min.features.get(feature)) /
```

```
(max.features.get(feature) -
min.features.get(feature));
            }
}

public class ListCalc {
            public static int sum (List<Integer> a){
      if (a.size() > 0) {
         int sum = 0;

         for (Integer i : a) {
            sum += i;
         }
         return sum;
      }
      return 0;
   }
   public static double mean (List<Integer> a){
      int sum = sum(a);
      double mean = 0;
      mean = sum / (a.size() * 1.0);
      return mean;
   }
   public static double median (List<Integer> a){
      int middle = a.size()/2;

      if (a.size() % 2 == 1) {
         return a.get(middle);
      } else {
        return (a.get(middle-1) + a.get(middle)) / 2.0;
      }
    }
}
```

```
ArrayList<Article> list_article =
ListBuilder.buildList(args[0]);
StylometryClassifier classifier = new
StylometryClassifier();


for(Article a: list_article){
            String path =
args[0]+"/"+a.getAuthor()+"/"+a.getCurrent().getNa
me();
            try {
                        classifier.addDocToClassifier(path,
a.getAuthor());
            } catch (Exception e) {
                        e.printStackTrace();
            }
}
```

ДОДАТОК Д АРХІТЕКТУРА КЛАСИФІКАТОРА



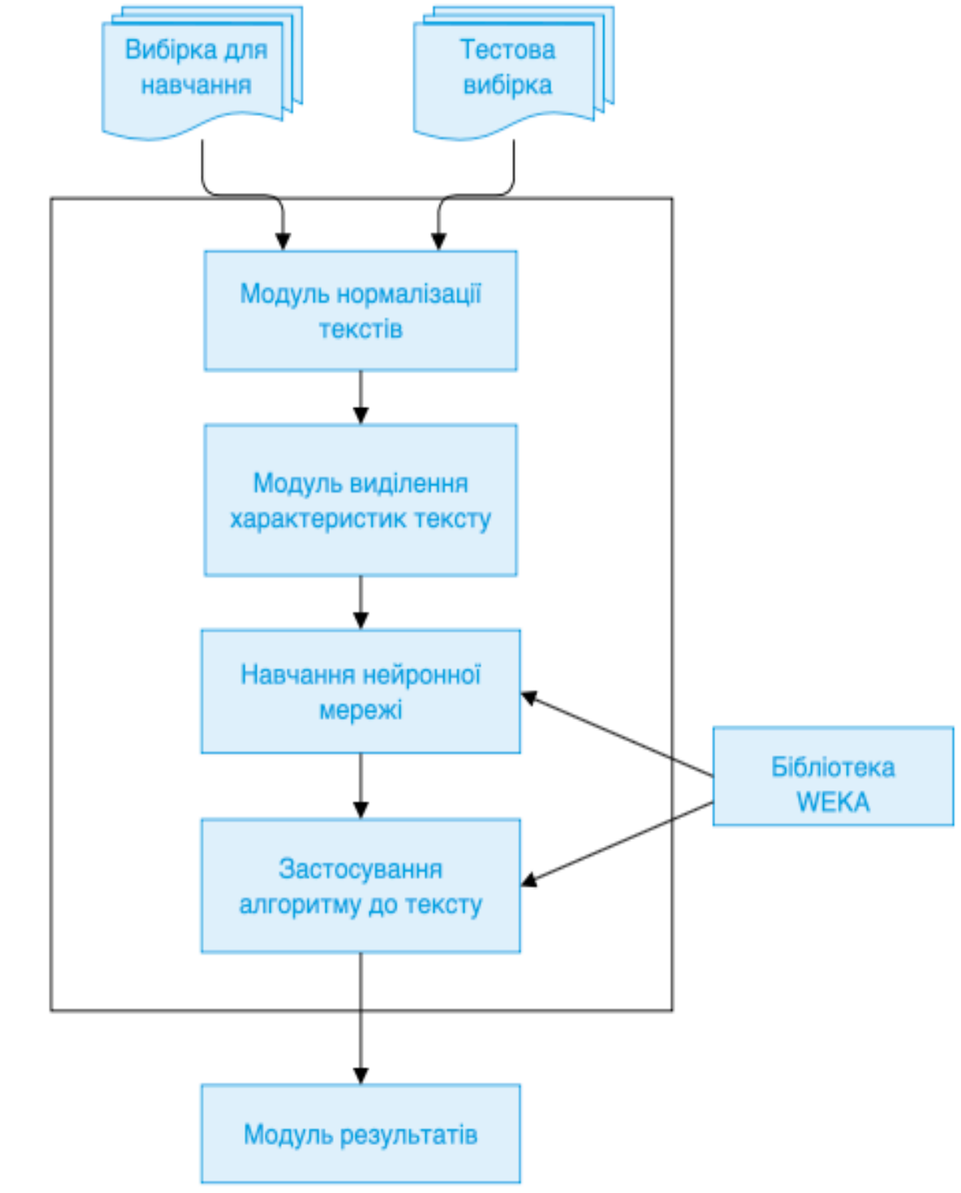
Демонстраційний плакат до магістерської дисертації

Архітектура кластфікатора

Виконав студент гр. ІП-82мп Храпов О.О

Керівник Фіногенов О.Д

ДОДАТОК Е СХЕМА РОБОТИ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ



Демонстраційний плакат до магістерської дисертації

Схема роботи програмного забезпечення

Виконав студент гр. ІП-82мп	Храпов О.О
Керівник	Фіногенов О.Д